Manual do Usuário

SPDS 6

Sistema para Programação, Documentação e Supervisão dos Controladores Industriais

Doc. JMU.0001 Versão 6.07 Setembro, 2001

Nota de Apresentação

Este manual foi elaborado pela HI Tecnologia Indústria e Comércio LTDA para documentação do Sistema de Programação, Documentação e Supervisão (SPDS) dos Micro Controladores Industriais modelos MCI02, MCI03 e ZAP500, projetados e desenvolvidos pela HI Tecnologia Ind. e Com. Ltda. Este ambiente juntamente com firmware próprio, configura os Micro Controladores Industriais HI como controladores programáveis, oferecendo recursos avançados para o desenvolvimento de sistemas de controle e para a supervisão de máquinas e processos industriais.

Em se tratando de um sistema de uso geral, a HI Tecnologia não se responsabiliza por produtos gerados a partir da utilização deste sistema, a menos que ela própria os tenha desenvolvido, e assim esteja registrado em contratos específicos de fornecimento. Este material não poderá ser reproduzido ou transmitido em nenhuma forma e por nenhum meio, eletrônico ou mecânico sem prévia autorização escrita da HI Tecnologia, detentora de todos os direitos deste sistema. Em casos de dúvida ou sugestões favor entrar em contato com :

HI Tecnologia
Indústria e Comércio LTDA

Av. Dr. Armando de Sales Oliveira , 445
13090-150 - CAMPINAS - SP

Fone: (0xx19) 3252-6044 (pabx)
Fax: (0xx19) 3251-4253

e-mail: hi@hitecnologia.com.br

Web Site: *www.hitecnologia.com.br*

Índice

1. Introdução 9
2. Apresentação do SPDS 10
2.1 Diagrama Funcional 10
2.1.1 Ambiente para Desenvolvimento de Programas 11
2.2 Requisitos para Operação 12
2.2.1 Ambiente de Desenvolvimento 12
2.2.2 Conteúdo do Disco do Ambiente de Desenvolvimento 12
2.2.3 Sistema Operacional e *Drivers* 13
2.3 Instalação do SPDS 13
2.3.1 Cópia de Segurança 13
2.3.2 Instalação do Ambiente de Desenvolvimento 13
3. A IHM do Ambiente de Desenvolvimento 15
3.1 Tela de Informações da Aplicação 15
3.2 Teclas de Funções 15
3.3 Menus 15
3.4 Janela de Auxílio (*Help*) 16
3.5 Telas de Entrada de Dados 16
3.6 Tela de Seleção de Arquivos 17
3.7 Janela de Mensagens 18
3.8 Janela de Erro / Advertência 18
3.9 Janela de Confirmação 19
4. Ambiente de Desenvolvimento 20
5. Projeto 21
5.1 Carrega Projeto 21
5.2 Projeto Novo 22
5.3 Elimina Projeto 23
6. Programação 24
6.1 Escolha do Controlador para a Aplicação 24
6.1.1 Configuração de *Hardware* do MCI-02 24
6.1.2 Configuração de *Hardware* do MCI-03 25
6.1.3 Configuração da Aplicação 26
7. Editor de Programas 28
7.1 Operadores 29
7.2 Comandos do editor 32
7.2.1 Inserção de elementos 33
7.2.2 Eliminação de um bloco - (DEL) 34
7.2.3 Auxílio ao usuário - (F1) 34
7.2.4 Localização de uma lógica - (F2) 34
7.2.5 Procura de um operador - (F3) 35
7.2.6 Identificação de um operador (F7/F4) 35
7.2.7 Documentação de uma Lógica - (F5) 35
7.2.8 Edição de operadores dos blocos - (F6) 37
7.2.9 Sair do editor salvando o programa - (F8) 37
7.2.10 Inserção de uma linha em branco - (ALT A) 37
7.2.11 Eliminação de uma linha - (ALT E) 37
7.2.12 Próximo operador disponível - (ALT D) 38
7.2.13 Cópia de áreas - (ALT I, F, L, C e R) 38
7.2.14 Especificação de mensagens da IHM - (ALT H) 38
7.2.15 Controle de Acesso ao Teclado 41
7.2.16 Programação de Senhas do Controlador 42
7.2.17 Programação de Mensagens de Eventos (ALT T) 43
7.2.18 Especificação de Constantes Reais 45
7.2.19 Alteração da base numérica (ALT B) 45
7.2.20 Troca de um contato [NA por NF] ou [NF por NA] (ALT N) 46
7.2.21 Salvar programa em disco - (ALT S) 46

7.2.22 Obter informações sobre elemento/bloco (ALT F1) 46
7.2.23 Sair do editor sem salvar o programa (ESC) 46
8. Diagramas *Ladder* 47
8.1 Conexões 51
8.1.1 Conexão Horizontal 51
8.1.2 Conexão Vertical 51
8.2 Contatos 52
8.2.1 Contato Normalmente Aberto (NA) 52
8.2.2 Contato Normalmente Fechado (NF) 53
8.3 Bobinas 53
8.3.1 Bobina Simples 54
8.3.2 Bobina *SET* 55
8.3.3 Bobina *RESET* 56
8.3.4 Bobina Pulso (*ONE SHOT*) 57
8.3.5 Bobina Oscilador 58
8.4 Movimentação de Dados 59
8.4.1 Movimentação Simples (MOV) 59
8.4.2 Movimentação Indexada (MVX) 61
8.4.3 Inicialização de Dados (FIL) 62
8.5 Temporizador (TMR) 65
8.6 Contadores 66
8.6.1 Contador Unidirecional (CNT) 66
8.6.2 Contador Bidirecional (UDC) 68
8.7 Funções Matemáticas 70
8.7.1 Operações Matemáticas (SUM, SUB, MUL, DIV) 70
8.7.2 Raiz Quadrada (SQR) 71
8.8 Deslocamento à Direita (SHR) 72
8.9 Deslocamento à Esquerda (SHL) 73
8.10 Comparações 75
8.10.1 Comparações Matemáticas (=, <>, <, <=, > e >=) 75
8.10.2 Teste Lógico *AND* 76
8.11 Operações Lógicas 77
8.11.1 Operador *AND* 77
8.11.2 Operador *OR* 78
8.11.3 Operador *XOR* 79
8.12 Blocos de Controle de Programa 81
8.12.1 Fim de Programa (ENDPG) 81
8.12.2 Inicio Relé Mestre / Fim de Relé Mestre 81
8.12.3 Bloco de Lógica (Subrotinas - BLK) 83
8.12.4 Início/Fim de Bloco de Lógica (Subrotinas - BEGBK e ENDBK) 84
8.12.5 Bobina de Salto (JUMP) 85
8.13 Interface Homem Máquina 86
8.13.1 Bloco de Sinalização de Eventos (EVENT) 86
8.14 *Blocos Especiais* 87
8.14.1 Relógio de Tempo Real (RTC) 87
8.14.2 Controle PID 88
8.14.3 Controle de Servo Motor (SEV) 91
8.14.4 Controle de Motor de Passo (Step Motor) 93
8.14.5 Gerador de Frequência (FQG) 95
8.14.6 Interface de Encoder (ECD) 97
8.14.7 Interface de Resolver (RSV) 98
8.14.8 Contador de Alta Velocidade (HSC) 100
8.14.9 Comunicação com PLC Remoto (RCB) 101
8.14.10 Configuração e Status do Controlador (CTS) 102
9. Controlador 104
9.1 Gerar Código para a Aplicação 104
9.2 Carrega Controlador 104
9.3 Ativa/Bloqueia Aplicação 105
9.4 Elimina Aplicação 105
9.5 Setup do Controlador 105

9.5.1 Inicializa NV-RAM 105
9.5.2 Ajusta Relógio/Calendário 106
9.5.3 Testa *Watch Dog Timer* 107
9.5.4 Configura Comunicação 107
9.5.5 Carrega Novo *Firmware* 108
9.6 Estatística da Aplicação 108
9.6.1 Mapa de Memória 108
9.6.2 Tempo de Ciclo 109
10. Depurador 110
10.1 Comandos de depuração 112
10.1.1 Auxílio ao usuário - (F1) 112
10.1.2 Procura Lógica - (F2) 112
10.1.3 Procura Operador (F3) 112
10.1.4 Procura próximo operador - (ALT P) 112
10.1.5 Consulta de Sinais - (F4) 113
10.1.6 Liga/Desliga Documentação das Lógicas (F5) 113
10.1.7 Altera o valor de uma variável (F6) 113
10.1.8 Programa máscara de retenção de sinais - (F7) 113
10.1.9 Liga/Desliga controlador - (F8) 114
10.1.10 Sair da depuração - (ESC) 114
10.2 Comandos para o MCI 114
10.2.1 Supervisão de Variáveis Analógicas 114
10.2.2 Supervisão de Variáveis Digitais 116
11. Documentação 118
11.1 Identificação de Sinais de Entrada/Saída 118
11.2 Identificação de Memórias/Contatos Auxiliares 119
11.3 Configuração do Controlador 119
11.4 Base Não Volátil 119
12. Impressão 120
12.1 Impressão da Documentação 120
12.2 Configuração da Impressora 122
13. Interface Homem Máquina 124
13.1.1 Configuração da função associada a uma tecla 124
14. Supervisão 126
14.1 Painel de Controle 126
14.2 *Trend* de variáveis 129
14.3 Base de Histórico 130

1. Introdução

Há alguns anos o computador passou a fazer parte de nossas vidas, e dia a dia, esta participação tem aumentado. A convivência com o computador tornou-se amplamente difundida, tanto em ambientes de trabalho, lazer, doméstico ou outros. Exemplos corriqueiros podem ser vistos nos bancos, nas lojas, na utilização de eletro-eletrônicos, tais como vídeo cassete, relógios, aparelhos de som, entre outros. Em todos estes casos, o computador nos presta algum tipo de serviço, a partir do acionamento de uma tecla, ou de uma seqüência delas.

Em alguma situações o computador está programado para a execução de serviços mais sofisticados, contando nestes casos, com o auxílio de especialistas que tenham formação na respectiva área do conhecimento humano.

Objetivando aproximar o usuário do computador, alguns programas foram e são desenvolvidos de forma que a aplicação incorpore o conhecimento do especialista. No caso específico da sub-área da Engenharia que engloba os tópicos de "Comandos elétricos e Controle", uma especialização do conhecimento através da qual o controle de máquinas e equipamentos é feito eletricamente, utilizou-se por muitos anos uma representação expressa através de diagrama de contatos de relés.

Neste tipo de aplicação, o computador substitui relés, temporizadores, controladores e demais dispositivos que possam ser representados de forma digital. Paralelamente a esta substituição, surgiram no mercado ferramentas de*software* dedicadas à elaboração das lógicas de controle então necessárias. Este tipo de programa herdou e expandiu a já difundida lógica de relés. As máquinas capazes de processar estes programas foram denominadas de CLP (controladores lógico programáveis), e nada mais são que computadores especializados neste tipo de aplicação. Para o projeto e a implementação de "Comandos elétricos e Controle" é utiliza-se freqüentemente os diagramas de contatos de relés, também conhecidos como Diagramas*Ladder*.

O sistema objeto deste manual é dedicado a este tipo de projeto e para tal disponibiliza vários recursos a fim de facilitar o trabalho do projetista. O**S**istema para **P**rogramação, **D**ocumentação e **S**upervisão dos Micro Controladores Industriais da HI, aqui denominado**SPDS**, permite ao usuário executar as seguintes tarefas:

- Programar a seqüência de lógicas de contatos de relés (etapa de edição);
- Transportar o programa final para um controlador dedicado (carga do programa);
- Testar a operação do programa aplicativo (depuração);
- Supervisionar o seu funcionamento do sistema (supervisão);
- Documentar o programa e o projeto (documentação).

Conceitualmente, o SPDS foi dividido em duas partes a saber:

- Ambiente de Desenvolvimento de Programas, aqui denominado ADP; e
- O Controlador industrial da HI que executa as lógicas, sendo o elemento de implementação do comando elétrico desejado, e aqui genericamente denominado de MCI

O ADP é um programa de fácil utilização, que pode ser executado em microcomputadores do tipo IBM-PC. Os controladores são os Micro Controladores Industriais da HI Tecnologia (MCI02, MCI03 e ZAP500) que fornecem várias opções de interfaceamento com processos a serem controlados.

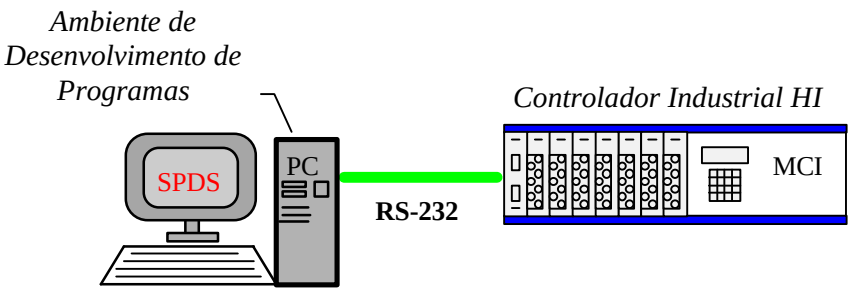
Nos próximos capítulos serão abordados vários itens relacionados aos microcontroladores industriais citados. O termo MCI, para microcontrolador industrial, será utilizado sempre que a referência for genérica.

2. Apresentação do SPDS

Neste capítulo são apresentados o diagrama funcional do Sistema para Programação, Documentação e Supervisão (SPDS) dos Micro Controladores Industriais HI, os requisitos de *hardware* e de *software* para sua operação, o conteúdo dos discos fornecidos, a descrição dos controladores associados, e os procedimentos para instalação do sistema. Recomendamos a leitura deste material antes de operar o sistema.

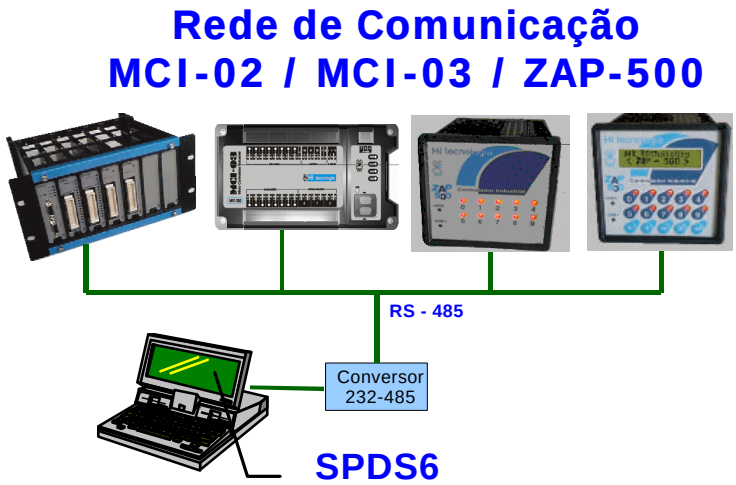
2.1 Diagrama Funcional

O SPDS é constituído por dois módulos interconectados, operando em equipamentos distintos. Conforme mostrado na figura abaixo, estes módulos são :



- Ambiente de Desenvolvimento de Programas (ADP)
- Micro Controlador Industrial da HI (MCI-02, MCI-03 ou ZAP-500)

Opcionalmente pode-se utilizar o SPDS em uma rede de comunicação (RS-485) conectados a vários equipamentos da família de controladores da HI Tecnologia, como ilustra a figura abaixo. Neste caso, cada controlador deve ser “endereçado” pelo SPDS através de um número de estação associado a cada controlador.



2.1.1 Ambiente para Desenvolvimento de Programas

O Ambiente para o Desenvolvimento de Programas foi concebido especialmente para facilitar a construção de programas para os controladores HI. Utiliza como plataforma de operação microcomputadores pessoais, do tipo IBM-PC, com sistemas operacionais, DOS, Windows (3.1, 95 ou NT). Sua interface é amigável, permitindo que sua operação seja feita através do teclado ou do *mouse.* Dispõe ainda de um sistema de auxílio contextual.

Através do Ambiente para Desenvolvimento de Programas (ADP) pode-se configurar e programar os controladores MCI's e, em uma fase posterior, depurar e supervisionar remotamente o programa desenvolvido. O sistema também provê todo o suporte necessário para a documentação, impressa ou em vídeo, do programa. Sugere-se que a documentação do programa seja feita durante a fase de programação.

A seguir faz-se uma descrição básica das funções disponibilizadas pelo ambiente de programação :

- **Configuração do Controlador Industrial** - Recurso utilizado para informar ao Ambiente para Desenvolvimento de Programas qual o *hardware* disponível para a programação. Através de um painel de configuração especifica-se, para cada *slot* (tanto do bastidor principal quanto do extendido - no caso do MCI02), o tipo de placa a ser usada. Deve-se especificar também se o sistema utilizará interface homem/máquina. A configuração deve preceder à etapa de programação.
- **Programação Ladder** - Recurso utilizado para a edição dos programas. Os programas são desenvolvidos através de um editor de diagramas de contatos dedicado (Diagramas *LADDER*). Neste módulo, além dos recursos para a edição do programa, também são disponibilizados recursos para a sua documentação.
- **Depuração (On-Line)** - Recurso utilizado para verificar se o programa está funcionando de acordo com o esperado. Durante esta etapa, a linha de comunicação serial é utilizada, interligando o MCI ao PC.
- **Documentação** - Recurso utilizado para a geração da documentação, impressa ou em vídeo, da aplicação. A documentação gerada consiste de um conjunto de relatórios contendo as seguintes informações: configuração do controlador; listagem do seu programa; descrição detalhada dos elementos utilizados no programa; tabela de referências cruzadas; uma descrição dos *straps* de configuração das placas utilizadas. Também é possível ao usuário configurar os parâmetros para a impressão do relatório.
- **Supervisão** - Recurso utilizado para supervisionar a operação do sistema. A supervisão consiste em se ter, na tela do PC, o valor das variáveis digitais e analógicas, e do bloco de controle **PID**, durante a execução do programa aplicativo.

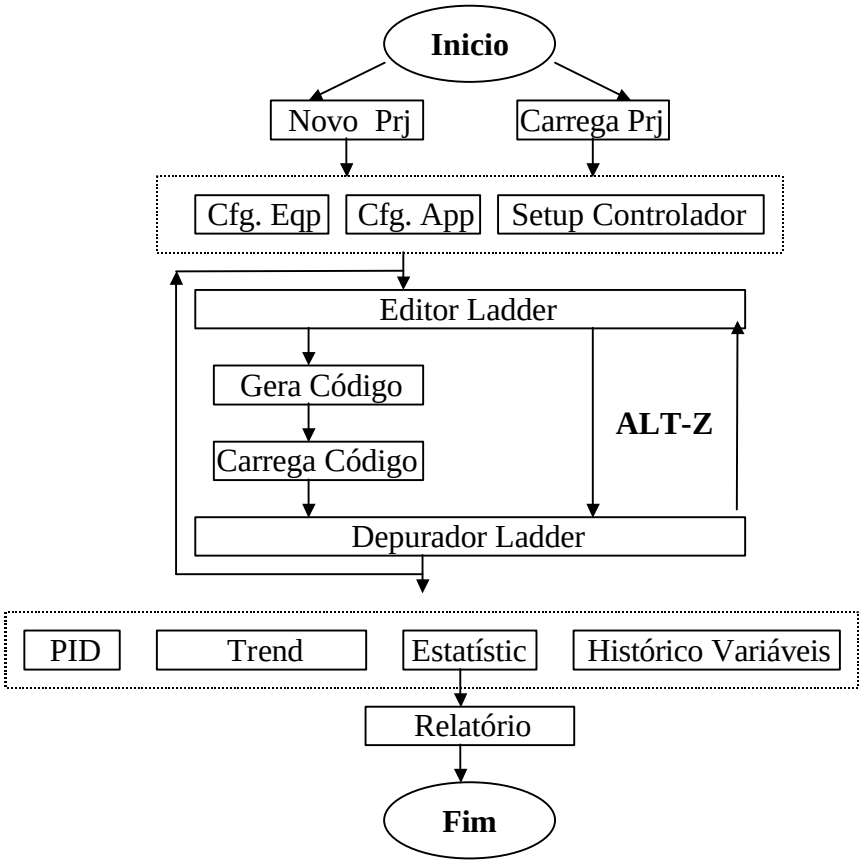


Figura - Diagrama Funcional do Ambiente SPDS

2.2 Requisitos para Operação

Neste item são apresentados os requisitos de*hardware* e *software* para que o SPDS possa ser utilizado.

2.2.1 Ambiente de Desenvolvimento

Este módulo foi projetado para rodar em microcomputadores do tipo IBM-PC, ou compatíveis, com uma configuração mínima de :

- 580 *Kbytes* de memória RAM livre para aplicação;
- 1 unidade de disco rígido de 1*Mbytes* livre para a aplicação;
- 1 unidade de disco flexível;
- 1 interface de vídeo preferencialmente VGA;
- 1 interface de comunicação serial (RS232);
- 1 sistema operacional MSDOS versão 3.1 ou Windows 3.1 ou Windows 95;
- 1 *mouse* Logitech ou compatível;
- 1 porta paralela padrão Centronics (para impressão de relatório);
- 1 impressora matricial, *laser* ou *Inkjet* (para impressão de relatório);

2.2.2 Conteúdo do Disco do Ambiente de Desenvolvimento

O disquete de instalação fornecido contém os seguintes arquivos :

SPDS40.EXE	Módulo executável do Ambiente de Desenvolvimento de Programas;
SPDSvr.C41	Arquivo contendo a configuração de cores utilizado pela interface homem/máquina do SPDS;
SPT0vr.T41 SPT1vr.T41	Arquivos contendo a configuração dos painéis de entradas de dados utilizado pela interface homem/máquina do SPDS;
SPDSVvr.P41	Arquivo contendo a de configuração de impressora utilizado pelo sistema de documentação do SPDS;
SPDSVvr.G41	Arquivo contendo a configuração das telas gráficas utilizadas pelo ambiente de supervisão do SPDS;
SPDSVvr.M41	Arquivo contendo a descrição da árvore de menus utilizado pela interface homem/máquina do SPDS;
SPH0Vvr.H41 SPH1Vvr.H41	Arquivos de auxílio (*help)* do sistema;
GUI41R3.T41	Arquivo contendo os painéis genéricos usados pela interface homem/máquina (GUI/Padrão);
MCI2Vvr.LIB	Arquivo que contém a biblioteca de funções do MCI02 para ser inserida em memória FLASH EPROM;
MCI3Vvr.LIB	Arquivo que contém a biblioteca de funções do MCI03 para ser inserida em memória FLASH EPROM;

Observação: Os arquivos especificados acima, possuem em seu nome as letras v e r, que representam a versão corrente (v) do arquivo e a sua revisão (r).

2.2.3 Sistema Operacional e *Drivers*

O SPDS foi desenvolvido para operar Sistemas Operacionais DOS versão 3.1 ou superior, podendo rodar sobre o Windows 3.1 ou Windows 95.

2.3 Instalação do SPDS

Pode-se realizar o download da versão corrente do SPDS no website da HI Tecnologia, cujo endereço é **www.hitecnol.com.br**.

2.3.1 Cópia de Segurança

Para se evitar transtornos causados por perdas acidentais dos arquivos originais. recomenda-se fazer uma cópia do disquete do sistema em outro local.

2.3.2 Instalação do Ambiente de Desenvolvimento

Para se instalar o ambiente deve-se seguir os seguintes passos :

- Executar o aplicativo SPDSxxx.EXE obtido através do download da Internet. Após a execução deste aplicativo serão extraídos um conjunto de arquivos associados a instalação do SPDS, dentre eles o programa de instalação SETUP.EXE.
- Executar o programa de instalação SETUP.EXE Após a execução inicia-se o processo efetivo de instalação do SPDS.

Este programa de instalação irá instalar o aplicativo SPDS no diretório de “Arquivos de programas\ HI Tecnologia\SPDS6”. Este diretório de instalação pode ser alterado pelo usuário durante o respectivo processo de instalação. Ao término da instalação é criado um ícone associado ao SPDS instalado. Para executar o SPDS basta clicar sob este ícone.

Observação : Certifique-se que o canal serial utilizado pelo *mouse* (no caso de *mouse* do tipo serial seja diferente do canal utilizado para comunicação com o controlador HI. Pode haver problemas de comunicação se o computador possuir algum *driver* instalado no canal serial utilizado para comunicação serial com o controlador.

2.3.2.1 Parâmetros opcionais de configuração

O SPDS está configurado para operar com monitor de vídeo colorido, e também para comunicar-se com o MCI através da porta serial COM1. Para a utilização de monitor de vídeo monocromático, e/ou para se utilizar a porta serial COM2, deve-se proceder da seguinte forma:

Para se especificar um monitor de vídeo monocromático, deve-se digitar :

SPDS6 b

Para se selecionar o canal de comunicação COM2, deve-se digitar :

SPDS6 2

ou opcionalmente, especificando as duas coisas :

SPDS6 b 2

3. A IHM do Ambiente de Desenvolvimento

Nesta seção são enfocados os aspectos básicos de operação e utilização oferecidos pelo sistema através de sua interface homem-máquina (IHM), padrão GUI/HI. As informações aqui expostas se aplicam a todos os contextos, e após sua leitura o usuário estará apto a "navegar" fluentemente pelo sistema, executando e experimentando todas suas funções, sem necessidade de consultar o manual para dirimir dúvidas relativas a IHM. O padrão IHM da HI Tecnologia faz uso extensivo de menus de opções, de teclas de funções, de janelas padronizadas (entre elas de informação, de entrada de dados, de auxílio (*help*), de confirmação, de advertência, de erro, etc.). Estas facilidades, além de permitirem a familiarização com o sistema em um curto espaço de tempo, também aumentam significativamente a produtividade do usuário, uma vez que os comandos podem ser rapidamente selecionados e executados, com uma probabilidade de incidência de erros extremamente baixa. A seguir são comentadas e exemplificadas as formas mais usadas de comunicação entre o sistema e o usuário.

3.1 Tela de Informações da Aplicação

Proj. Progr. Control. Depur. Superv. Estat. Docum. Impr. Info Fim
SPDS
Micro Cont. Indust.
Versão 4 - Rev. 00 - Liberação: Nao Lib.
Propriedade
HI Tecnologia - Beta Teste Final
Desenvolvimento
Versao em Desenvolvimento
Data e Hora Correntes
11/NOV/1994 - 08:47:51 Hs.
Dir corrente : C:\S40DEMO
Memória (RAM) Disponível : 82544 Bytes
Pressione ESC p/ continuar
F1_Auxilio | Projeto = [ ] | Memoria Disponivel: 82544 Bytes

Em todas as aplicações desenvolvidas no padrão de interface Homem-Máquina GUI existe uma tela de informações que é apresentada ao usuário logo no início da operação do ambiente. Esta tela apresenta informações gerais tais como, nome do sistema, versão, revisão, data de liberação, propriedade, etc. Esta tela tem o objetivo de identificar o sistema e pode ser acessada a qualquer instante através do menu de operação. Para prosseguir pressione **ESC**.

3.2 Teclas de Funções

As teclas **F1**, **F2**, ..., **F10**, referenciadas genericamente por "teclas de funções" são bastante utilizadas ao longo do sistema. Sempre que uma tecla de função estiver habilitada, serão apresentadas no vídeo informações sobre sua função naquela instância. Normalmente a tecla **F1** é reservada para acesso ao sistema de auxílio da aplicação. A ação associada a uma tecla de função é executada tão logo ela seja pressionada.

3.3 Menus

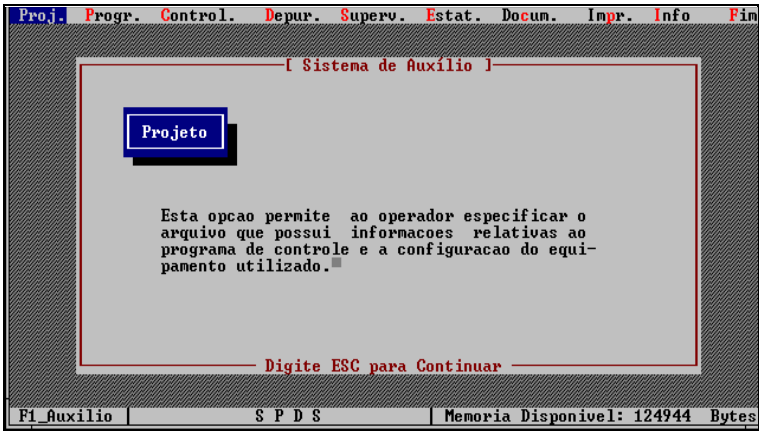
Forma clássica de se escolher uma das opções em um conjunto disponível. São utilizados dentro do sistema, dois tipos de menus - o horizontal (régua de opções ou menu principal) e o vertical (ativado a partir de uma das opções do menu principal). Em qualquer destes menus, uma das opções estará sempre em vídeo reverso, indicando que ela é a opção corrente. Para se ativar uma determinada opção, deve-se proceder da seguinte forma : Selecionar a opção desejada através das teclas de seta (horizontais p/ menu horizontal e verticais p/ menu vertical) e em seguida pressionar **ENTER**. Em todas as opções, um dos caracteres aparece em destaque (cor diferente), facilitando a identificação da tecla de seleção direta. Existem dois casos a considerar :

- **Menus verticais** - A opção desejada pode ser ativada pressionando-se simplesmente a tecla correspondente ao caracter em destaque ou ativando-se a opção através das setas para baixo e para cima, ou ainda pressionar a opção desejada através do *mouse*.

- **Menus horizontais** - A opção desejada pode ser ativada pressionando-se simultaneamente a tecla **ALT** e a tecla correspondente ao caracter em destaque. Se o *mouse* estiver habilitado, pode-se selecionar a opção desejada posicionando seu cursor sobre a área associada, pressionando-se em seguida o seu botão esquerdo. Outra maneira de ativar uma opção no menu horizontal é mover o cursor do menu até a opção desejada e em seguida pressionar a tecla de seta para baixo. Para cancelar um determinado "sub-menu", e retornar ao nível anterior, basta pressionar a tecla **ESC** (escape) ou o botão direito do *mouse* (se habilitado).

Obs: Os menus somente estarão habilitados, se nenhum tipo de tela tiver sido empilhada no vídeo, após sua respectiva abertura.

3.4 Janela de Auxílio (*Help*)


Proj. Progr. Control. Depur. Superv. Estat. Docum. Impr. Info Fim
[Sistema de Auxílio]
Projeto
Esta opcao permite ao operador especificar o arquivo que possui informacoes relativas ao programa de controle e a configuracao do equipamento utilizado.
Digite ESC para Continuar
F1_Auxilio
S P D S
Memoria Disponivel: 124944 Bytes

Na maioria das vezes, durante a operação do sistema, existe uma tecla de função (normalmente a tecla **F1**) associada a função de auxílio (*help*). Ao se pressionar esta tecla será aberta uma janela, contendo informações específicas para aquela instância do programa. Assim, por exemplo, se o usuário estiver preenchendo um determinado campo de entrada de dados, e tiver qualquer tipo de dúvida acerca do dado a ser inserido, poderá pressionar a tecla de auxílio, e o sistema lhe fornecerá as informações cabíveis naquela instância. Uma janela de auxílio pode conter mais de uma página de informação. Neste caso o usuário poderá acessar as páginas através das teclas de **PgUp** e **PgDn**.

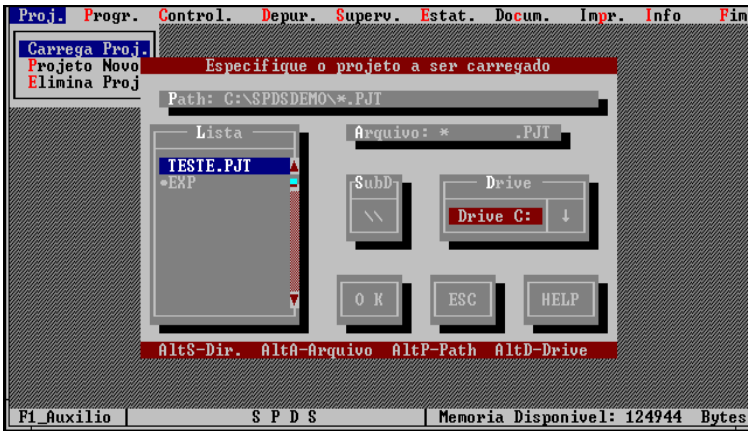
3.5 Telas de Entrada de Dados

As telas de entradas de dados, como o próprio nome revela, são janelas utilizadas pelo sistema, nas situações em que há interação com o usuário. Estas telas são compostas por um conjunto de campos de entradas de dados, compartilhando características, tais como :

- Todos os campos são apresentados, quando possível, com um valor pré-definido (*default*);
- O campo de entrada corrente é apresentado em vídeo reverso;

- O valor pré-definido de um campo, poderá ser recuperado pressionando-se a tecla **ESC** (escape), desde que o campo não tenha sido validado;
- Os valores numéricos são submetidos a uma pré-validação, antes de se validar o campo correspondente. Caso o valor esteja fora da faixa prevista para aquele campo, o valor pré-definido será restaurado, e o valor anterior desprezado;
- Se o *mouse* estiver habilitado, qualquer campo poderá ser diretamente selecionado por ele.
- Em caso de se digitar caracteres errados, pode-se utilizar livremente a tecla ← (*backspace)*, que elimina o último caracter digitado e reposiciona o cursor na posição correta.
- Em algumas situações, principalmente para entrada de seqüências de caracteres, quando o campo de entrada de dados é menor que a seqüência a ser lida, o sistema provê o recurso do "*scrolling*", que mostra apenas os últimos caracteres da seqüência inserida.
- Dados do tipo "enumerado", são entrados da seguinte forma : selecione o campo desejado (ele apresentará um de seus valores possíveis). Para alterá-lo pressione "espaço" ou **Tab** até que o valor apresentado coincida com o valor desejado. Observe que "espaço" altera as opções no sentido inverso de **Tab**. Este tipo de dado pode também ser alterado pressionando-se o botão esquerdo do *mouse* por duas vezes consecutivas. Observe também que a lista de opções é circular.
- Existem campos cujo dado associado pode assumir apenas dois valores distintos por exemplo, Sim / Não, Ligado/Desligado etc. Para alterar estes campos utilize a tecla de espaço ou pressione o *mouse* no campo desejado.
- Para se alterar o sinal do valor de um campo numérico, basta pressionar a tecla **-**", qualquer que tenha sido o número de caracteres entrados, a recomposição do campo é automática. Na entrada de dados do tipo "real" os caracteres **.**" e "**,**" são utilizados indistintamente. A validação ou cancelamento de uma tela de entrada de dados é efetuada na maioria dos casos pelas teclas **ENTER** e **ESC** respectivamente.

3.6 Tela de Seleção de Arquivos



Quando o programa de aplicação necessitar que o usuário especifique o nome de um arquivo, será apresentada uma tela de seleção de arquivos. Nesta tela são apresentadas todas as informações necessárias para especificação de um nome de arquivo.

Um menu apresenta os nomes dos arquivos que preenchem a especificação contida no campo diretório. Caso o nome do arquivo a ser selecionado esteja presente na lista basta selecioná-lo através das teclas de setas ou do *mouse*. Caso seja necessário alterar o diretório ou disco corrente deve-se proceder da seguinte maneira.

Através da tecla **TAB** selecione o campo de diretório, em seguida especifique o novo diretório, ou diretamente o *driver* a ser acessado. No caso de arquivo novo, especifique diretamente o nome no campo arquivo.

- O campo "\" permite que o sistema procure o(s) arquivo(s) especificado(s) no diretório imediatamente superior ao corrente.
- Valide a entrada de dados através da tecla **ENTER** ou cancele através da tecla **ESC**.

Estão disponíveis para o usuário os seguintes comandos:

- **ALT P -** Posiciona o cursor no campo *'Path*" da tela permitindo que ele seja modificado.
- **ALT L -** Posiciona o cursor na lista de arquivos permitindo que se selecione um arquivo qualquer constante na lista;
- **ALT S -** Permite que o usuário mude de diretório;
- **ALT D -** Permite que o usuário mude de *drive*, podendo especificar qualquer *drive* existente em seu computador;
- **ALT A -** Posiciona o cursor no campo "Arquivo" da tela permitindo que ele seja modificado;

3.7 Janela de Mensagens

Janelas de mensagens são utilizadas, normalmente, para informar ao usuário que uma determinada ação (comando) foi ou está sendo executado. Um exemplo típico de utilização desta janela é aberta pelo sistema para informar ao usuário que um determinado arquivo de dados foi, ou está sendo gravado em disco. Esta janela normalmente aparece no vídeo durante um curto espaço de tempo (tipicamente 3 segundos) ou até que o usuário pressione a tecla ESC.

3.8 Janela de Erro / Advertência

Proj. Progr. Control. Depur. Superv. Estat. Docum. Impr. Info Fim
Mapa de Memoria
Tempo de Ciclo
Falha no acesso aos arquivos de trabalho
ESC-Continua
F1_Auxilio
S P D S
Memoria Disponivel: 124944 Bytes

"Janelas de Erro" e "Janelas de Advertência" são utilizadas sempre que alguma situação indesejável ocorre. Assim, por exemplo, ao se tentar ler um arquivo inexistente, ou quando não for possível alocar a quantidade de memória necessária para um dado aplicativo, estas janelas serão abertas contendo as mensagens necessárias, para que o usuário possa resolver o problema e continuar a operar o sistema.

A figura abaixo ilustra uma janela de erro típica. A diferença básica entre uma janela de erro e uma de informação, além da cor, é quanto a gravidade do erro. Janelas de erro são utilizadas em situações "graves" que impedem a continuação da execução do programa, se o erro levantado não for corrigido. A janela de advertência é retirada do vídeo automaticamente após decorrido um período de tempo. Já a janela de erro só será eliminada do vídeo através da ação do usuário pressionando a tecla **ESC**.

3.9 Janela de Confirmação

Para reduzir o risco de ativar acidentalmente algum comando "desastroso", o sistema solicita a confirmação do usuário, sempre que a circunstância o exigir. Assim, comandos dos tipos "Eliminar arquivos", "encerrar sessão", "interromper processamento", etc., serão sempre precedidos de uma tela de confirmação, com a mensagem adequada. A figura a abaixo apresenta uma Janela de Confirmação típica.

O texto apresentado na parte superior da janela especifica a função (comando) a ser confirmado. Retângulos delimitam as palavras "SIM" , "NÃO", ("CANCELA"). Um dos retângulos estará sempre ativo, e será ativado caso a tecla **ENTER** seja pressionada. A opção selecionada pode ser escolhida de uma das seguintes maneiras :

- Utilizando-se uma das teclas de setas horizontais, pode-se selecionar a opção desejada, e então pressionar a tecla **ENTER**.
- Pressionando-se o caracter em destaque da opção desejada.
- Posicionando o *mouse* na seleção desejada e pressionando-se seu botão direito. Uma vez escolhida a opção desejada, a tela de confirmação é retirada do vídeo e o comando associado é executado.

4. Ambiente de Desenvolvimento

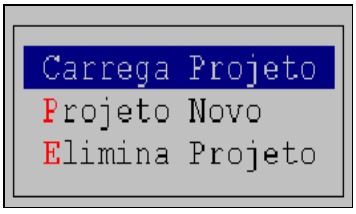
O Ambiente de Desenvolvimento de Programas do SPDS oferece ao usuário vários recursos para elaboração, documentação e desenvolvimento de um projeto. No SPDS, a idéia de um projeto consiste na aglutinação sob um mesmo nome (Projeto) , de todos os seus elementos constituintes, ou seja, programa, documentação, código gerado para os micro controladores MCI02 e MCI03 e informações sobre a configuração do sistema. Antes do usuário executar qualquer operação, é necessário que se carregue um projeto e se este não existir, deve-se criá-lo. Os recursos disponíveis ao usuário podem ser acessados através da tela principal do SPDS, na qual existem, através em um menu as seguintes opções :

- **Projeto** - esta opção permite ao usuário Carregar um Projeto ("Carrega Proj.") existente em disco, criar um Novo Projeto ("Projeto Novo") ou Eliminar um projeto ("Elimina Proj.") existente em disco.
- **Programa** - esta opção permite ao usuário editar um programa utilizando-se de um editor de diagramas de contato ("Editor *LADDER*"), escolher o equipamento no qual desenvolverá a sua aplicação ("Equipamento"), podendo ser MCI02 ("Equipamento" + "Control. MCI02") ou MCI03 ("Equipamento" + "Control. MCI03") e ainda, configurar o controlador para uma determinada aplicação ("Conf. Aplicação").
- **Controlador** - esta opção permite ao usuário gerar código executável ("Gera Código Aplic." + "Modo de Desenvolvimento") e enviá-lo para o MCI ("Carrega Controlador"). Uma vez carregado o programa no micro controlador MCI, pode-se ativar o controlador ("Ativa/Bloq. Aplicação") ou interrompê-lo ("Ativa/Bloq. Aplicação"). Após a validação do programa gerado, têm-se a opção de gerar código para ser colocado diretamente em memória do tipo FLASH EPROM no micro controlador MCI ("Gera Código Aplic." + "Modo de Aplicação"). Está também disponível, através da opção "*Setup* Controlador" , a possibilidade do usuário ter acesso à base de dados não volátil dos micro controladores MCI02 ou MCI03 ("*Setup* Controlador" + "Inicializa NV-RAM"). Pode-se ler a base da dados não volátil do micro controlador MCI e transferi-la para disco ("*Setup* Controlador" + "Inicializa NV-RAM" + "NV-RAM→Disco"), editá-la ("*Setup* Controlador" + "Inicializa NV-RAM" + "Edita NV-RAM") e enviá-la de volta ao micro controlador MCI ("*Setup* Controlador" + "Inicializa NV-RAM" + "Disco→NV-RAM"). Existe ainda a possibilidade de zerar-se a base de dados não volátil do micro controlador MCI diretamente através da opção "*Setup* Controlador" + "Inicializa NV-RAM" + "Inic.NV-RAM". Adicionalmente, pode-se programar a data e hora correntes do relógio calendário do micro controlador MCI ("*Setup* Controlador" + "Ajusta Relógio/Calend."). Pode-se também testar o *Watch Dog Timer* dos micro controladores através da opção ("*Setup* Controlador" + "Testa *Watch Dog Timer*") e escolher qual o canal de comunicação será utilizado para a comunicação entre o ADP e o micro controlador MCI ("*Setup* Controlador" + "Conf. Comunicação"). Através da opção ("Estatística Aplicação"), pode-se obter um mapa de memória do micro controlador MCI02 ou MCI03 para o programa gerado, para memória RAM, NVRAM e EPROM ("Estatística Aplicação" + "Mapa de Memória") e obter informação informações sobre o tempo de ciclo de um programa ("Estatística Aplicação" + "Tempo de Ciclo").
- **Depurador** - esta opção permite ao usuário visualizar, através de uma tela similar à existente no editor de diagramas *Ladder*, o seu programa em funcionamento no micro controlador MCI. Pode-se visualizar, durante a execução do programa no micro controlador MCI, os estados dos contatos, memórias e bobinas de qualquer lógica existente no programa ("Depurador"). Esta ferramenta é de grande ajuda no desenvolvimento de programas, já que torna possível a localização de problemas durante a sua execução.
- **Supervisão** - esta opção permite ao usuário realizar dois tipos de supervisão, a de blocos de controle (PID) existentes em seu programa ("Painel de Controle") e gráficos de tendência de variáveis analógicas ("*Trend* de Variáveis"). A supervisão de blocos de controle é configurável e pode-se supervisionar até quatro blocos simultaneamente na tela. Na opção de gráficos de tendência pode-se supervisionar até quatro variáveis simultâneas em dois gráficos distintos, apresentando duas variáveis por gráfico.

- **Relatório** - esta opção permite ao usuário gerar uma documentação completa do projeto que está desenvolvendo, em Vídeo ("Em Vídeo") ou Impressora ("Impressora"). Através dela, pode-se fazer: uma especificação dos sinais utilizados pelo programa("Em Vídeo" + "Id. dos Sinais de I/O"), uma especificação das memórias/contatos auxiliares utilizados no programa("Em Vídeo" + "Ident. Memórias/Contatos"), gerar uma documentação da configuração do MCI02 ou MCI03 ("Em Vídeo" + "Conf. do Controlador") e gerar um mapa de memória da base não volátil ("Em Vídeo" + "Base Não Volátil"). A especificação de sinais pode ser feita para as entradas digitais ("Em Vídeo" + "Id. dos Sinais de I/O" + "Entradas Digitais"), saídas digitais ("Em Vídeo" + "Id. dos Sinais de I/O" + "Saídas Digitais"), entradas analógicas ("Em Vídeo" + "Id. dos Sinais de I/O" + "Entradas Analógicas") e saídas analógicas ("Em Vídeo" + "Id. dos Sinais de I/O" + "Saídas Analógicas"). A especificação de memórias/contatos pode ser feita para contatos auxiliares ("Em Vídeo" + "Ident. Memórias/Contatos" + "Contatos Auxiliares"), memórias inteiras ("Em Vídeo" + "Ident. Memórias/Contatos" + "Memórias Inteiras") e reais ("Em Vídeo" + "Ident. Memórias/Contatos" + "Memórias Reais"). O mapa da base não volátil pode ser gerado para contatos auxiliares ("Em Vídeo" + "Base Não Volátil" + "Contatos Auxiliares"), memórias inteiras ("Em Vídeo" + "Base Não Volátil" + "Memórias Inteiras") e reais ("Em Vídeo" + "Base Não Volátil" + "Memórias Reais"). A opção ("Impressora" + "Conf. Documento") permite configurar a impressão da documentação a ser gerada, imprimir a documentação gerada ("Impressora" + "Imprime Documento"), com possibilidade de ajuste da impressora/relatório ("Impressora" + "Config. Impressora").
- **Info (Informação)** - esta opção permite ao usuário visualizar algumas informações a respeito do SPDS, tais como : a identificação do sistema, versão, revisão e a data de sua liberação, têm-se a propriedade do SPDS e a identificação de quem o desenvolveu. Nesta tela é apresentado a data e hora correntes, o diretório corrente e a memória total disponível para o usuário.
- **Fim** - esta opção permite ao usuário sair do sistema. Antes é pedido uma confirmação desta decisão.

Nos capítulos seguintes, são apresentadas todas os recursos disponíveis no SPDS para o usuário de uma forma mais detalhada.

5. Projeto



No SPDS considera-se que uma aplicação a ser desenvolvida pelo usuário estará associada a um projeto. Para iniciar o desenvolvimento de uma aplicação, o usuário deve criar um projeto. Um projeto no SPDS aglutina sob um mesmo nome (nome do projeto) vários tipos de informações pertinentes ao mesmo. Toda informação gerada pelo sistema para um determinado projeto é salva em disco.

O sub-menu de "Projeto" coloca disponível para o usuário as seguintes opções : "Carrega Projeto", "Projeto Novo" e "Elimina Projeto". Estas três opções serão apresentadas a seguir.

5.1 Carrega Projeto

Quando o usuário executa o SPDS, ele verifica se o programa foi rodado anteriormente e se havia carregado algum projeto. Se esta condição for satisfeita, o SPDS automaticamente carrega este projeto. Se esta condição não for satisfeita, ele automaticamente seleciona a opção "Carrega Projeto" do menu "Projeto". Ao ser selecionada esta opção é aberta uma janela de especificação do nome do projeto. Neste caso o usuário deve selecionar o projeto que deseja carregar e tecla *Return* ou clicar com o botão esquerdo do *mouse* no campo identificado "OK". Procedendo desta forma o projeto será carregado e o programa estará apto a operar.

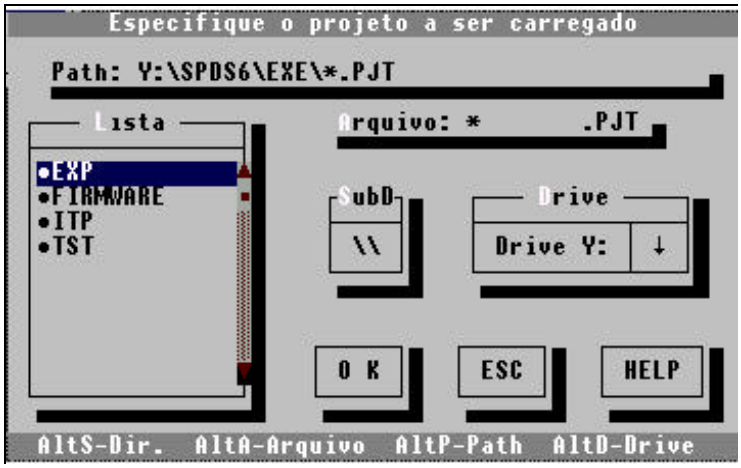


Figura - Tela de especificação do nome do projeto a ser carregado.

Existe uma linha na parte de baixo da janela principal do SPDS. Nesta linha estão várias informações de status. Entre elas está o nome do projeto corrente, juntamente com o tipo do controlador que está sendo utilizado. A parte referente ao projeto na linha de status é mostrada abaixo.

Figura - Indicação do nome do projeto corrente na linha de status do SPDS

5.2 Projeto Novo

Através da opção “Projeto Novo” do menu “Projeto”, pode-se criar um novo projeto no SPDS. Ao se selecionar esta opção é aberta uma janela de especificação do nome do projeto. Esta tela é equivalente à que é mostrada através da opção “Carrega Projeto”. Ao fornecer um nome para o projeto ele será criado caso não exista.

Um projeto no SPDS aglutina sob um mesmo nome (nome do projeto) vários tipos de informações pertinentes ao mesmo. Toda informação gerada pelo sistema para um determinado projeto é salva em disco. Ao se criar um projeto novo, está-se dizendo ao SPDS para salvar estas informações em arquivos com o nome do projeto. Em um projeto podemos ter várias informações gravadas em disco. Para que um tipo de informação se distinga de outra é utilizada uma extensão diferente para cada arquivo gerado.

Podemos ter para um projeto as seguintes extensões:

PJT - identifica um arquivo contendo informações de gerenciamento do projeto e a configuração de *hardware* do MCI.

LAD - identifica um arquivo que contém o fonte do diagrama de contatos (*Ladder*).

LDD - identifica um arquivo que contém os identificadores (comentários) de início de cada lógica do programa *Ladder*.

BSI - identifica um arquivo que contém os identificadores dos sinais dos módulos de *hardware* do MCI (entradas/saídas digitais/analógicas).

RSI - identifica um arquivo que contém os identificadores dos contatos auxiliares existentes no programa *Ladder*.

MSI - identifica um arquivo que contém os identificadores das memórias inteiras existentes no programa *Ladder*.

DSI - identifica um arquivo que contém os identificadores das memórias reais existentes no programa *Ladder*.

PFX - identifica um arquivo que contém as lógicas definidas no programa do usuário em uma notação pós-fixada. Este arquivo é utilizado para a geração de código executável para o MCI.

LOG - identifica um arquivo que contém os identificadores de lógica do programa *Ladder*. Através deste arquivo, o sistema identifica o início e fim de cada lógica.

NVB - identifica um arquivo que contém uma cópia da base de dados não volátil existente no MCI . Esta base contém os contatos auxiliares, memórias inteiras e reais alocados na memória não volátil do MCI (NVRAM).

Todos os arquivos descritos acima são gerados e gerenciados automaticamente pelo SPDS e o usuário tem acesso a eles apenas através de sua interface. Todos esses arquivos são binários com um formato desconhecido para o usuário. Se através do DOS, o usuário mandar imprimir o seu conteúdo (usando o comando *type*, por exemplo), aparecerá no vídeo/impressora a seguinte mensagem:

SPDS V6 Rxx
<mensagem de identificação do arquivo>

Onde:

SPDS - identifica o arquivo como pertencente à aplicação do SPDS,
V6 - identifica a versão atual do SPDS (seis),
Rxx - indica a revisão atual do SPDS.
<mensagem de identificação do arquivo>, identifica sucintamente o tipo do arquivo e pode ser:

- **Ident. Sinais Mem. Reais**, para os arquivos com extensão DSI.
- **Ident. Sinais Mem. Int,** para os arquivos com extensão MSI.
- **Ident. Sinais Placas**, para os arquivos com extensão BSI.
- **Ident. Sinais Contatos**, para os arquivos com extensão RSI.
- **Prog. Aplicação - MCI**, para os arquivos com extensão PJT.
- **Base de Prog. Ladder**, para os arquivos com extensão LAD.
- **Doc. Programa Ladder**, para os arquivos com extensão LDD.
- **Arq. de Equações**, para os arquivos com extensão PFX.
- **Arq. Ident. Lógicas**, para os arquivos com extensão LOG.
- **Arq. Base Não Volátil**, para os arquivos com extensão NVB.

Obs: O usuário não deve jamais alterar o conteúdo dos mesmos, já que as consequências podem ser imprevisíveis.

5.3 Elimina Projeto

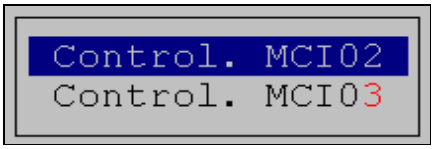
Através da opção “Elimina Projeto” do menu “Projeto”, pode-se eliminar um projeto no SPDS. Ao se selecionar esta opção é aberta uma janela de especificação do nome do projeto. Esta tela é equivalente à que é mostrada através da opção “Carrega Projeto”. Ao fornecer um nome para o projeto ele será eliminado.

6. Programação

A programação dos micro controladores industriais (MCI) é feita no Ambiente de Desenvolvimento de Programas através de um editor de diagramas de contatos (diagramas *Ladder*). Antes de começar a programar, deve-se especificar qual a configuração de *hardware* disponível no MCI.

No menu de Programa (Programa) estão disponíveis recursos que permitem : a edição de diagramas de contato ("Editor Ladder"), escolher o micro controlador e configurar o seu *hardware* ("Equipamento" + "Control. MCI02") ou ("Equipamento" + "Control. MCI03"), a aplicação ("Conf. Aplicação"). A seguir serão apresentados estes recursos de programação, na ordem que devem ser acessados durante a criação de um projeto.

6.1 Escolha do Controlador para a Aplicação



Antes de se iniciar a edição de um programa usando o editor de diagramas *Ladder*, deve-se especificar em qual equipamento ela será utilizada. Para isto existe a opção "Controlador", que permite especificar que será utilizado o equipamento MCI-02, MCI-03 ou ZAP-500. Uma vez escolhido o controlador, no caso do MCI-02 e MCI-03, deve-se especificar os respectivos módulos de *hardware* a serem utilizados pela aplicação. Para o controlador ZAP-500 assumi-se que existe o módulo de expansão ZEM-530.

6.1.1 Configuração de *Hardware* do MCI-02

Antes de se iniciar a edição de um programa usando o editor de diagramas *Ladder*, deve-se informar ao sistema qual é a configuração dos módulos disponível no MCI02. Para isto existe a opção ("Controlador" + "Control. MCI02") no menu de Programa ("Programa"). Ao selecionar esta opção é aberta uma janela de configuração, onde é mostrado um MCI02 com os seus dois sub-bastidores, principal e extendido. Em cada um deles são mostrados os *slots* disponíveis, ou seja, 8 (oito) *slots* para o principal e 14 (quatorze) para o extendido.

Os *slots* são numerados de 00 a 21, perfazendo um total de 22 *slots* no MCI02 em sua configuração máxima. No sub-bastidor principal o *slot* de número 00 é reservado para colocação de uma CPU (unidade central de processamento) e o de número 07 é reservado para a placa de controle da interface de teclado/*display* do MCI02 (KBD), quando utilizada. Se esta placa não for utilizada, pode-se colocar qualquer outra em seu lugar, caso seja necessário. À medida que o *hardware* vai sendo configurado, é atualizado na tela, em campos correspondentes e individualmente, o número total de pontos de Entrada/Saída Analógicos/Digitais.

De acordo com as necessidades do usuário, vai-se selecionando que placas irão em quais *slots*. Para executar a configuração de placas de I/O do MCI02, dispõe-se das seguintes teclas :

- ↓ → ↑ ← - permitem a seleção do *slot* a ser configurado;
- **espaço** - permite selecionar o módulo (placa) que ocupará o *slot*;
- **RET** - confirma a configuração definida pelo usuário;
- **ESC** - cancela a configuração. Neste caso, se já havia uma configuração anterior, ela continuará inalterada;

Configuracao dos Slots do M C I
08 09 10 11 12 13 14 15 16 17 18 19 20 21
00 01 02 03 04 05 06 07
M C I 0 2
[ ] I H M
Remota
No de Canais Disponiveis
Ent. Digitais
Ent. Analogicas
Sai. Digitais
Sai. Analogicas
RET-Confirma
ESC-Cancela
↓ → ↑ ← - Seleciona Slot
Espaco - Seleciona Modulo

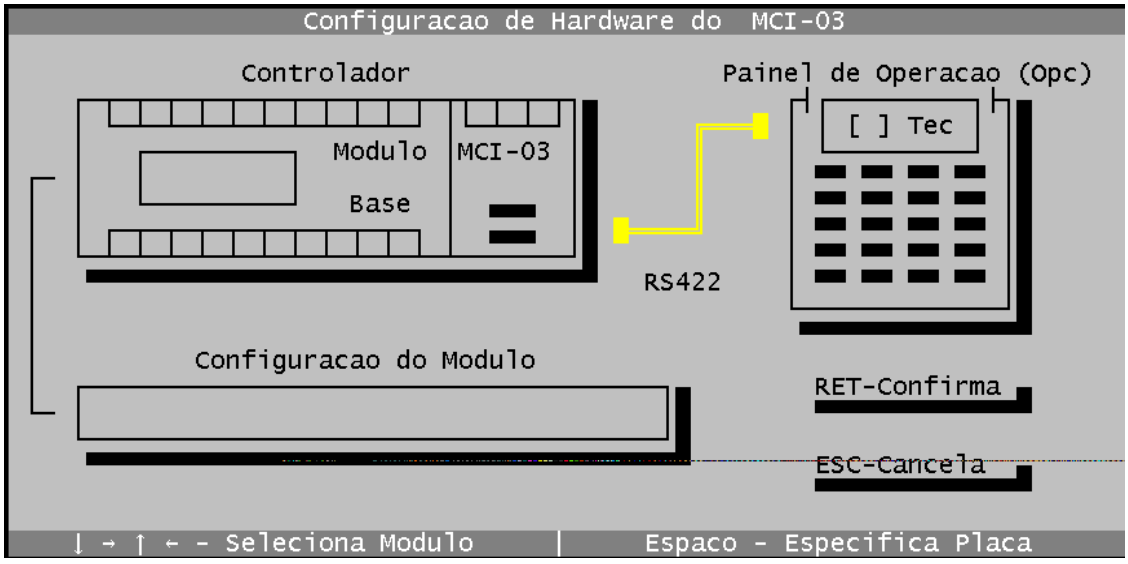
É importante observar que uma vez feita esta configuração, ela poderá ser utilizada como referência para a aquisição das placas que comporão a aplicação no MCI02, bastando para isto tirar uma listagem através da opção de documentação do SPDS, onde consta uma lista detalhada da configuração de *hardware* do MCI02.

6.1.2 Configuração de *Hardware* do MCI-03

Antes de se iniciar a edição de um programa usando o editor de diagramas *Ladder*, deve-se informar ao sistema qual é a configuração dos módulos disponível no MCI03. Para isto existe a opção (“Controlador” + “Control. MCI03”) no menu de Programa ("Programa"). Ao selecionar esta opção é aberta uma janela de configuração, onde é mostrado um MCI03.

De acordo com as necessidades do usuário, vai-se selecionando que módulos serão utilizados. Para executar a configuração do MCI03, dispõe-se das seguintes teclas :

- ↓ → ↑ ← - permitem a seleção do módulo a ser configurado;
- **espaço** - permite selecionar o módulo que será utilizada;
- **RET** - confirma a configuração definida pelo usuário;
- **ESC** - cancela a configuração. Neste caso, se já havia uma configuração anterior, ela continuará inalterada;



É importante observar que uma vez feita esta configuração, ela poderá ser utilizada como referência para a aquisição das placas que comporão a aplicação no MCI03, bastando para isto tirar uma listagem através da opção de documentação do SPDS, onde consta uma lista detalhada da configuração de *hardware* do MCI03.

6.1.3 Configuração da Aplicação

O controlador possui alguns parâmetros que devem ser configurados pelo usuário para a sua aplicação. Esses parâmetros dizem respeito ao controlador, à interface Homem/Máquina e à sua Base Não Volátil. Este parâmetros serão apresentados a seguir.

Base Não Volátil

Os parâmetros específicos deste item definem qual a quantidade de memória retentiva que será reservada para o programa do usuário. Memória retentiva é aquela que quando a energia do controlador é desligada, o conteúdo da memória não é perdido, ao contrário da não retentiva que perde o seu conteúdo.

Este recurso está disponível ao usuário através da opção "Conf. Aplicação" do menu de programação ("Programa"). Estes parâmetros, apresentados a seguir, estão separados funcionalmente para maior clareza :

- **Mem. Inteiras e Mem. Reais -** este item especifica o número máximo de posições de memória retentiva (NMaxMemRet) que deverão ser utilizadas pelo usuário. Uma vez selecionado este valor, está-se definindo automaticamente que as memórias retentivas serão numeradas de 0 a este valor menos um (NMaxMemRet -1) e que as memórias não retentivas serão numeradas deste valor em diante. Por exemplo, se o valor definido pelo usuário for 2, as memórias retentivas serão as de número 0 e 1 (M0000 e M0001 ou D000 e D0001) e as não retentivas terão o seu valor inicial igual a 2 (M0002 ou D0002 até o máximo utilizado pelo usuário).
- **Contatos -** este item especifica o número máximo de contatos auxiliares retentivos (NMaxContRet) que deverão ser utilizadas pelo usuário. Uma vez selecionado este valor, está-se definindo automaticamente que os contatos auxiliares retentivos serão numerados de 0 a este valor menos um (NMaxContRet -1) e que os contatos auxiliares não retentivos serão numerados deste valor em diante. Por exemplo, se o valor definido pelo usuário for 5, os contatos auxiliares retentivos serão os de número 0, 1, 2, 3 e 4 (R0000, R0001,

R0002, R0003 e R0004) e os não retentivos terão o seu valor inicial igual a 5 (R0005 até o máximo utilizado pelo usuário).

Controlador

- **Power-up -** este item permite ao usuário informar ao controlador que o seu programa, deverá ser executado automaticamente ou não, ao ser ligado a energia do MCI. Se o usuário especificar uma Execução Automática (“Auto”), o programa será executado automaticamente ao ser ligado o MCI e caso contrário, ficará parado aguardando um comando do usuário. Este comando de execução pode ser feito através da interface do SPDS, sub-menu "Ativa/Bloq. Aplicação" do menu "Controlador", ou através da interface homem/máquina do MCI. Se o MCI não estiver configurado com a interface homem/máquina, a única opção possível é a execução automática da aplicação. Maiores detalhes sobre a ativação do programa via operador podem ser encontrados no capítulo "Interface Homem/Máquina do MCI" neste manual, no item "Ativação da Aplicação no MCI via Operador".
- **Resol. Timer -** este item permite ao usuário especificar a resolução do bloco temporizador do MCI. O usuário tem a possibilidade de especificar a seguintes resoluções: 100, 80, 60, 40 e 20 milisegundos. Este parâmetro determina qual o período de tempo em que os blocos de *timers* existentes no programa serão verificados para ver se alcançaram o final da contagem. Quanto menor este parâmetro, mais rapidamente os blocos de *timers* serão verificados, entretanto a base de tempo do *timer*, que é de 0.1 segundos, permanece inalterada. Este parâmetro não serve para alterar a base mínima de tempo dos blocos de *timers* mas para aumentar a sua precisão.
- **Aqu. Analógica -** este item permite ao usuário especificar o período da aquisição/atuação analógica do MCI (tempo entre duas aquisições/atuações de entradas/saídas analógicas). O usuário tem a possibilidade de especificar os seguintes períodos: 200, 100, 80, 60, 40, 20 e Mínima. Neste caso cada aquisição/atuação ocorrerá a cada 0.2, 0.1, 0.08, 0.06, 0.04 , 0.02 segundos ou no menor tempo que for possível (opção Mínima).

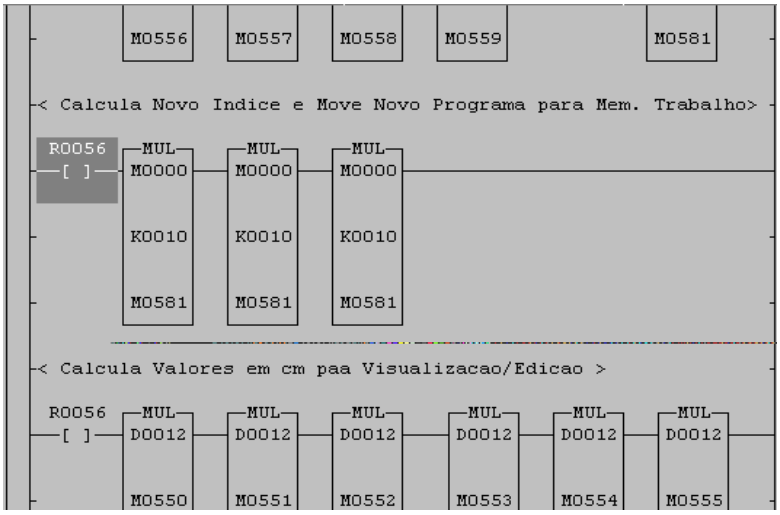
IHM (interface homem/máquina)

- **Atualização IHM -** este item permite ao usuário especificar o período de atualização do *display* da IHM do MCI (intervalo de tempo entre duas atualizações do *display*). O usuário tem a possibilidade de especificar os seguintes períodos: 2, 1, 0.7, 0.5 e 0.3 segundos.
- **Precisão Msg Evento -** este item permite ao usuário especificar a precisão no caso de se mostrar no *display* da IHM um número real associado a uma mensagem de evento. O usuário tem a possibilidade de especificar a seguintes precisões: Máxima, Sem Precisão, uma casa após a vírgula (X.X), duas casas (X.XX), três casas (X.XXX), quatro casas (X.XXXX), cinco casas (X.XXXXX) e seis casas (X.XXXXXX).
- **Precisão Msg Progr -** este item permite ao usuário especificar a precisão no caso de se mostrar no *display* da IHM um número real. O usuário tem a possibilidade de especificar a seguintes precisões: Máxima, Sem Precisão, uma casa após a vírgula (X.X), duas casas (X.XX), três casas (X.XXX), quatro casas (X.XXXX), cinco casas (X.XXXXX) e seis casas (X.XXXXXX).

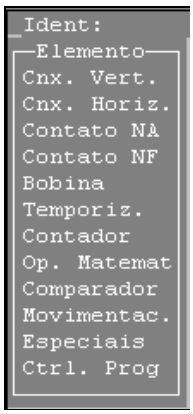
7. Editor de Programas

Para se editar um programa de controle para ser executado no MCI, o SPDS dispõe de um editor de programas *Ladder*. Ao selecionar a opção "Editor Ladder" no menu de Programa ("Programa"), é aberta uma tela do editor de diagramas*Ladder*. Nesta tela existem alguns campos que têm uma função específica que será apresentada a seguir juntamente com uma descrição do funcionamento do editor.

- **Campo de Edição** - este campo é utilizado para a edição do programa do usuário em diagrama *Ladder*. É composto por uma matriz de 8 linhas por 8 colunas, onde pode-se inserir ou eliminar os elementos de controle das lógicas desejadas. Neste campo existe um cursor (quadrado apresentado em vídeo reverso), que serve como referência para os comandos existentes no editor. Como um programa pode possuir várias lógicas e estas serem maiores do que o espaço reservado para mostrá-las no vídeo, pode-se utilizar as teclas**PgDn, PgUp, Home** e **End** para andar pelo programa. As teclas de seta são utilizadas para mover o cursor dentro do campo de edição.



- **Campo de Elementos** - este campo apresenta os elementos disponíveis para inserção no programa (diagrama de contatos). Nele estão disponíveis 11 elementos que podem ser acessados simplesmente pressionando-se a tecla correspondente à letra que estiver diferenciada no nome do elemento, por exemplo, para se inserir na janela na posição do cursor um elemento representando um contato normalmente aberto, deve-se pressionar a tecla **A** (Contato NA), já que é este caracter que está diferenciado neste elemento. Este tipo de seleção é válido para qualquer dos comandos existentes nesta janela. Alguns ao serem selecionados inserem diretamente o elemento na posição do cursor na janela de edição, enquanto outros oferecem algumas opções que devem ser selecionadas pelo usuário, como por exemplo, ao se pressionar a tecla**C** (Contador), abre-se um menu com as seguintes opções : Contador Unidirecional e Contador Bidirecional. Neste menu o usuário escolhe o elemento que deseja inserir no programa. Todos os elementos existentes nesta janela e os respectivos elementos associados às opções, serão apresentados mais detalhadamente neste manual.

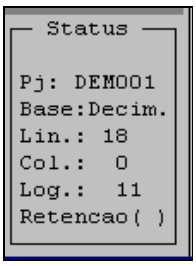


- **Campo de Status** - este campo tem como objetivo ajudar o usuário a se orientar durante a edição de uma lógica. Nele é mostrado a linha, coluna e lógica correntes correspondendo à posição onde se encontra o cursor no campo de edição. Ao se iniciar a edição de um novo programa, o marcador do número de lógica estará igual a zero durante toda a edição. Somente

após uma análise sintática do diagrama feito pelo usuário é que este marcador estará indicando o valor correto da lógica onde estiver localizado o cursor. Isto acontece pois somente após a análise sintática é que o sistema saberá onde começam e terminam as lógicas do programa. A análise é feita toda vez que se pressiona o comando que salva o programa em disco (**ALT S**) ou o que deixa a edição ("F8 - Fim"). Se o usuário estiver editando um programa anteriormente editado, este marcador estará mostrando corretamente a lógica corrente.

Neste campo também é indicado para o usuário, se um operador que está apontado pelo cursor é ou não retentivo. Existe neste campo um texto que indica o estado de retenção do operador que é o seguinte: Retenção ( ). Se o operador não for retentivo, o texto estará apresentado como mostrado anteriormente e aparecerá, diferenciado se o mesmo for retentivo, da seguinte maneira: Retenção (✓).

- **Campo de Projeto** - este campo tem como objetivo informar ao usuário qual o projeto em que se está trabalhando. No SPDS, todos os arquivos gerados pelo sistema possuem o nome do projeto, variando apenas a sua extensão em função de sua utilização.



- **Campo de Documentação** - neste campo é mostrado o identificador de cada elemento por onde estiver passando o cursor na janela de edição. Este campo fica aberto durante toda a edição, possibilitando ao usuário identificar qualquer operador que estiver presente no programa.

- **Campo de Comandos** - este campo mostra ao usuário alguns dos recursos disponíveis durante a edição do programa. Estes recursos são acessados via as teclas de função, de F1 a F8. Para selecioná-los é necessário apenas que se pressione a tecla de função correspondente ao recurso desejado. Estes recursos são:

  F1 - Help - permite obter auxílio durante a edição do programa;
  F2 - Lógica - permite localizar alguma lógica dentro do programa;
  F3 - Procura - permite procurar por um operador durante a edição do programa;
  F4 - Identif. - permite inserir/alterar o identificador de uma lógica;
  F5 - Docum. - permite inserir/alterar o identificador de um operador qualquer;
  F6 - Edita - permite alterar um operador;
  F7 - Nomeia - permite inserir/alterar o identificador de um operador;
  F8 - Fim - permite sair do editor e checar a sintaxe do programa;

- **Campo de Base** - este campo mostra ao usuário o tipo de base que está sendo utilizada no editor, que pode ser decimal ou hexadecimal.

7.1 Operadores

Os recursos de interface do MCI (sinais digitais e analógicos) com o processo a ser controlado são identificados por operadores utilizados no programa de controle. A tabela a seguir apresenta todos os operadores utilizados pelo SPDS, juntamente com a identificação do tipo e a faixa permitida de valores (*range)* de cada um.

Operador	Descrição	Quantidade	"*Range* Oper."	"*Range* Conteúdo"
I	Entrada Digital	312 *	0..311 máx.	0/1 [*OFF*/*ON*]
O	Saída Digital	312 *	0..311 máx.	0/1 [*OFF*/*ON*]
E	Entr. Analóg.	168 *	0..167 máx	0..4095
S	Saída Analóg.	84 *	0..83 máx	0..4095
K	Const. inteira	-	-	-999.. +9999
H	Const. Hexadecimal	-	-	0..FFFF
M	Memór. inteira	500	0..499	-32768..+32767
Q	Constante real	100	0..99	-10E-38...+10E38
D	Memória real	150	0..149	-10E-38 .. +10E38
R	Cont. auxiliar	1200	0 .. 1199	0/1[*OFF*/*ON*]
T *1	Lógica	-	2..Lógicas-1	-
T *2	Mens. Sinal. Eventos	150	0 .. 149	-
T *3	Contr. PID	4	0..3 máx	-

Tabela de descrição de Operadores

* - A quantidade destes operadores em uma aplicação depende da configuração de *hardware* do MCI02 (número de módulos de entrada/saída digitais e analógicos).

*1, *2, *3 - T é um operador genérico e é utilizado por vários blocos. No caso da tabela acima, *1 é utilizado com o bloco de salto, *2 é utilizado com o bloco de sinalização de eventos, *3 é utilizado com o bloco de controle PID.

Estes operadores são utilizados nos blocos de controle durante a confecção de um programa e para tal é necessário que se faça um maior detalhamento da função de cada um no contexto do editor de diagramas de contatos.

I - operador que representa uma entrada digital do MCI. Como no MCI podemos ter várias entradas digitais é necessário distingui-las. Isto é possível através de uma numeração sequencial dos operadores. Esta numeração inicia-se com 0 (zero) e seu valor final depende do número de placas de entradas digitais existentes no MCI. Por exemplo, I0000 representa o canal 0 (zero) da primeira placa de entradas digitais e I0015, corresponde ao canal 15 desta mesma placa. No item "Documentação dos Sinais" do menu de Relatório é possível identificar os sinais associados a cada placa de entrada presente no MCI.

O - operador que representa uma saída digital do MCI. Como no MCI podemos ter várias saídas digitais é necessário distingui-las. Isto é possível através de uma numeração sequencial dos operadores. Esta numeração inicia-se com 0 (zero) e seu valor final depende do número de placas de saídas digitais existentes no MCI. Por exemplo, O0000 é o canal 0 (zero) da primeira placa de saída digital e O0015, corresponde ao canal 15 desta mesma placa;

R - operador que representa um contato auxiliar de um relé. Estes são contatos sem conexão direta com o meio físico (processo) e são úteis na definição das lógicas. Como podemos ter vários contatos auxiliares em um relé é necessário distingui-los. Isto é possível através de uma numeração sequencial dos operadores. Esta numeração inicia-se com 0 (zero) e seu valor final depende do número de contatos utilizados no programa do usuário. Um limitante superior é a quantidade de memória disponível para alocação destes contatos. Convencionou-se neste sistema que se o usuário especificar a existência de contatos auxiliares retentivos, todos os contatos a partir de zero até o número especificado - 1 serão retentivos e neste caso os seguintes serão não retentivos (veja item "Configuração do Controlador" para maiores detalhes);

M - operador que representa um valor inteiro de 16 bits armazenado na memória do MCI (-32768 a +32767), podendo guardar valores pertinentes aos vários elementos de uma lógica. Por exemplo, em um bloco Temporizador, o valor total da temporização pode estar em uma posição de memória. As memórias são usadas nas funções de entrada e saída analógicas para guardar os valores lidos/escritos das placas analógicas do MCI. Como podemos ter várias memórias é necessário distingui-las. Isto é possível através de uma numeração sequencial dos operadores. Esta numeração inicia-se com 0 (zero) e seu valor final depende da quantidade de memória utilizadas pelo usuário em seu programa. Um limitante superior é a quantidade de memória disponível para alocação destas memórias. Convencionou-se neste sistema que se o usuário especificar a existência de memórias retentivas, todas as memórias a partir de zero até o número especificado - 1 serão retentivas e neste caso as seguintes serão não retentivas (veja item "Configuração do Controlador" para maiores detalhes);

D - operador que representa um valor real armazenado na memória do MCI (-10E-38 a 10E+38), podendo guardar valores pertinentes aos vários elementos de uma lógica. Como podemos ter várias memórias reais é necessário distingui-las. Isto é possível através de uma numeração sequencial dos operadores. Esta numeração inicia-se com 0 (zero) e seu valor final depende da quantidade de memórias utilizadas pelo usuário em seu programa. Um limitante superior é a quantidade de memória disponível para alocação destas memórias. Convencionou-se neste sistema que se o usuário especificar a existência de memórias retentivas, todas as memórias a partir de zero até o número especificado - 1 serão retentivas e neste caso as seguintes serão não retentivas (veja item "Configuração do Controlador" para maiores detalhes);

K - operador que representa o valor de uma constante inteira de 16 bits. Este valor pode ser usado por vários elementos de uma lógica e em várias aplicações. Por exemplo, em um elemento Contador o valor de uma constante pode ser usado para determinar o valor total da contagem. Como visto na tabela, uma constante pode assumir valores na faixa de -999 a 9999; O valor da constante é fornecido diretamente no operador.

H - operador que representa o valor de uma constante hexadecimal. Este valor pode ser usado por vários elementos de uma lógica e em várias aplicações. Por exemplo, em um elemento Contador o valor de uma constante pode ser usado para determinar o valor total da contagem. Como visto na tabela, uma constante pode assumir valores na faixa de 0 a FFFF; O valor da constante é fornecido diretamente no operador.

Q - operador que representa o valor de uma constante real. Este valor pode ser usado por vários elementos de uma lógica e em várias aplicações. Como visto na tabela, uma constante pode assumir valores na faixa de -10E-38 a 10E+38; O usuário especifíca no programa uma constante e o seu valor é inserido através de uma tabela. No item 7.2.18 “Especificações de Constantes Reais”.

E - operador que representa um canal de Entrada Analógica. Este valor é usado pelo bloco de Movimentação, MOV. Por exemplo, quando se deseja ler o valor de uma entrada analógica do MCI em uma memória, insere-se na lógica um bloco funcional MOV, cujo operador E especificará o canal correspondente para a leitura do sinal do processo e o segundo operador, M, indicará a memória para a qual será transferido o valor da entrada analógica. O número associado a este operador, pode variar de 0 até o número máximo de entradas analógicas disponíveis no MCI - 1.

S - operador que representa um canal de Saída Analógica. Este valor é usado pelo bloco de Movimentação, MOV. Por exemplo, quando se deseja escrever um valor em uma saída

analógica do MCI, insere-se na lógica um bloco funcional MOV, cujo segundo operador, S ,especificará o canal correspondente para a escrita do sinal do processo. O número associado a este operador, pode variar de 0 até o número máximo de saídas analógicas disponíveis no MCI - 1.

T - operador que representa uma malha de controle PID. Este valor é usado pelo bloco de Controle PID. Por exemplo, quando se deseja ter uma malha de controle PID no programa, insere-se na lógica um bloco funcional PID, cujo operador T especificará o bloco utilizado. O número associado a este operador, pode variar de 0 até 3, correspondendo ao número máximo de malhas de controle PID disponíveis para utilização MCI.

T - operador que identifica uma lógica qualquer presente em um programa. Este valor é usado pelo bloco de salto, que tem como função saltar para a lógica especificada pelo operador T caso a entrada da bobina seja energizada. O número associado a este operador, pode variar do número da lógica em que está o bloco de salto + 1 até o número máximo de lógicas existentes no programa.

T - operador que representa um elemento de sinalização de eventos. Este valor é usado pelo bloco de sinalização de eventos, para identificar qual é o número da mensagem de sinalização de evento a ser mostrada no*display* do MCI, ao ter a sua entrada energizada. O número associado a este operador, pode variar de 0 até o número máximo de mensagens de sinalização de eventos especificadas pelo usuário.

Estes operadores podem estar presentes em vários blocos de uma lógica. Alguns blocos possuem dois ou três operadores. O que determina qual operador estará presente em um bloco é a definição do próprio bloco e em alguns casos, se o bloco assim o permitir, a escolha do usuário.

7.2 Comandos do editor

Neste item são descritos os comandos disponíveis no editor de diagramas de contatos. Estes comandos têm como objetivo facilitar o máximo possível a edição do programa do usuário.

Existem comandos disponíveis ao usuário através: de teclas de função, da combinação de algumas teclas com a tecla **ALT**, das teclas **INS** e **ESC** e os disponíveis diretamente através de letras associadas ao menu de inserção de elementos.

Os comandos disponíveis ao usuário através das teclas de função são:

F1 - Help - permite obter auxílio durante a edição do programa;
F2 - Lógica - permite localizar alguma lógica dentro do programa;
F3 - Procura - permite procurar por um operador durante a edição do programa;
F4 - Identif. - permite inserir/alterar o identificador de uma lógica;
F5 - Docum. - permite inserir/alterar o identificador de um operador qualquer;
F6 - Edita - permite alterar um operador;
F7 - Nomeia - permite inserir/alterar o identificador de um operador;
F8 - Fim - permite sair do editor e checar a sintaxe do programa;

Os comandos disponíveis através da combinação de algumas teclas com a tecla**ALT**, são:

- **ALT A**
  Permite inserir uma nova linha na tela do editor;
- **ALT E**
  Permite eliminar uma linha na tela do editor;

- **ALT D**
  Permite visualizar quais os próximos operadores disponíveis para utilização;
- **ALT I**
  Permite marcar o início de uma área da tela do editor para posterior cópia;
- **ALT F**
  Permite marcar o fim de uma área da tela do editor para posterior cópia;
- **ALT G**
  Permite localizar uma determinada lógica dentro do programa do usuário. Tem-se o mesmo efeito teclando-se o comando "F2-Lógica";
- **ALT L**
  Permite copiar a área da tela marcada para um *buffer* temporário;
- **ALT C**
  Permite copiar a área que estiver no *buffer* temporário para a posição do cursor;
- **ALT R**
  Permite retirar a marca da área que estiver marcada na tela do editor;
- **ALT H**
  Permite ao usuário especificar vários parâmetros da interface homem/máquina do MCI02;
- **ALT Q**
  Permite ao usuário especificar constantes Reais para posterior utilização no seu programa;
- **ALT B**
  Permite ao usuário especificar a base que deseja trabalhar, decimal ou hexadecimal;
- **ALT N**
  Permite ao usuário alterar um contato NA por um NF e vice e versa;
- **ALT P**
  Permite ao usuário procurar o próximo operador especificado através do comando "F3-Procura";
- **ALT S**
  Permite ao usuário salvar o projeto que estiver trabalhando no momento;
- **ALT T**
  Permite programar parâmetros extendidos de operadores ou blocos;
- **ALT F1**
  Permite ao usuário obter informações sobre um bloco/elemento específico. Para isto deve-se posicionar o cursor sobre o elemento/bloco que deseja e teclar simultaneamente as teclas **ALT** e **F1**;

O comando disponível através da tecla **ESC** é apresentado a seguir:

ESC - Fim - permite sair do editor sem salvar as alterações feitas no programa;

Nos itens a seguir serão apresentados estes comandos, começando pelos de inserção e eliminação de elementos, seguidos dos comandos disponíveis através das teclas de função **INS** e posteriormente os disponíveis através da combinação de algumas teclas com a tecla **ALT**.

7.2.1 Inserção de elementos

A inserção de elementos no campo de edição é feita com o auxílio do campo de elementos. Neste campo estão apresentados os elementos disponíveis para inserção no programa (diagrama de contatos). Nela estão disponíveis 11 elementos que podem ser acessados simplesmente pressionando-se a tecla correspondente à letra que estiver diferenciada no nome do elemento, por exemplo, para se inserir no campo na posição do cursor um bloco representando um contato

normalmente aberto, deve-se pressionar a tecla **A** (Contato NA), já que é este caracter que está diferenciado neste bloco. Este tipo de seleção é válido para qualquer dos elementos existentes neste campo. Alguns ao serem selecionados inserem diretamente o elemento na posição do cursor no campo de edição, enquanto outros oferecem algumas opções que devem ser selecionadas pelo usuário, como por exemplo, ao se pressionar a tecla **C** (Contador), abre-se um menu com as seguintes opções : Contador Unidirecional e Contador Bidirecional. Neste menu o usuário escolhe o bloco que deseja inserir no programa. Para se inserir qualquer bloco é necessário que na posição do cursor não haja qualquer outro, ou seja, esta posição deve estar livre. Os blocos funcionais necessitam ainda que haja um espaço vertical livre equivalente ao dobro ou triplo do cursor, dependendo do bloco. Durante a inserção de um bloco é necessário que se especifique qual o operador que será utilizado. Esta especificação inclui o tipo e o número do operador, como por exemplo I0032. Alguns blocos necessitam de dois ou mais operadores. A inserção de um bloco só se completará com a especificação de todos os seus operadores.

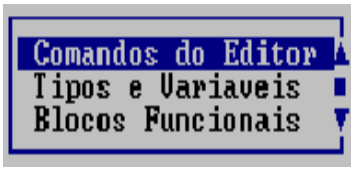
7.2.1.1 Tecla Insert (INS)

Através desta tecla pode-se inserir elementos/blocos entre elementos/blocos de uma lógica qualquer. Ao ser pressionada, é aberto um espaço horizontal entre os elementos, na posição do cursor, suficiente para a inserção do elemento/bloco que se deseja.

7.2.2 Eliminação de um bloco - (DEL)

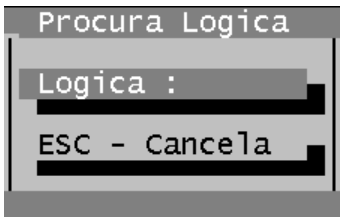
Para se eliminar um bloco qualquer, basta que o cursor seja posicionado sobre ele e se pressione a tecla **DEL**. Se o bloco for representado por mais de um elemento é necessário que o cursor esteja posicionado na parte superior do bloco para que este comando seja executado.

7.2.3 Auxílio ao usuário - (F1)



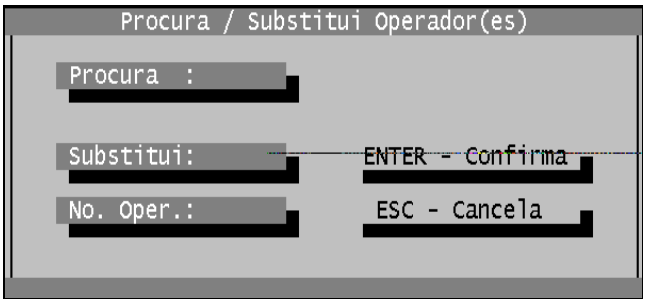
A qualquer instante durante a edição de programas é possível ao usuário, obter informações sobre a edição do programa. Ao se pressionar a tecla **F1**, abre-se um menu com as seguintes opções: "Comandos do Editor", "Tipos e Variáveis" e "Blocos Funcionais". O usuário deve escolher um dos itens que desejar para que o sistema o informe, através de uma janela de auxílio, os comandos e as informações pertinentes sobre o item selecionado. Este comando aparece no campo de comandos como "F1-Auxílio".

7.2.4 Localização de uma lógica - (F2)



Durante a edição de um programa às vezes se faz necessário localizar uma lógica qualquer existente no programa. Para localizar uma lógica qualquer no programa deve-se pressionar a tecla **F2**, ou opcionalmente, simultaneamente as teclas **ALT** e **G**. Ao se pressionar esta tecla, abre-se uma janela onde aparece um espaço para ser preenchido com o número da lógica que se deseja localizar. Após o preenchimento deste valor, o editor localiza a lógica e a coloca no campo de edição. Deve-se observar que este comando só localizará uma determinada lógica se ela existir e se o programa tiver sido salvo em disco pelo menos uma vez após a edição da lógica (comando **ALT S** ou "F8-Fim"). Se a lógica for encontrada o cursor será posicionado na primeira coluna da primeira linha da lógica. Este comando aparece no campo de comandos como "F2-Lógica".

7.2.5 Procura de um operador - (F3)



Durante a edição de um programa às vezes se faz necessário localizar um determinado operador. Ao se pressionar **F3**, abre-se uma janela onde aparece um espaço para o usuário especificar qual o operador que deseja localizar. Se o operador existir, o mesmo será localizado e o cursor ficará indicando a sua posição. Se houver alguma outra ocorrência do operador no programa, pode-se teclar simultaneamente as teclas **ALT** e **P**, que a próxima ocorrência do operador será automaticamente localizada. Desta forma, para localizar todas as ocorrências de um determinado operador, basta ir teclando**ALT** e **P**, que os mesmos aparecerão na ordem em que estiverem no programa. Este comando aparece na janela de comandos como "F3-Procura".

7.2.6 Identificação de um operador (F7/F4)

Em um programa escrito através de um diagrama de contatos existem vários blocos representando grandezas físicas que fazem parte do processo que se deseja controlar, como por exemplo um contato normalmente aberto pode representar um fim de curso de uma máquina ou a chave de acionamento de um motor. Para que o programa possa ser lido e entendido com mais facilidade pelo usuário, permite-se que para cada operador no programa exista um identificador associado. Este identificador deve o mais fielmente retratar a natureza do operador na aplicação para facilitar a leitura do programa. Através da tecla**F7** pode-se inserir/alterar o identificador do elemento que estiver indicado pelo cursor. Este comando aparece no campo de auxílio como "F7Nomeia".Alternativamente pode-se especificar o identificador de todos os operadores de um programa simultaneamente. Esta especificação é feita separadamente por classes de operadores. Ao teclar-se "F4-Identif.", abre-se um menu com sete opções onde o usuário pode escolher entre : Entradas Digitais, Saídas Digitais, Entradas Analógicas, Saídas Analógicas, Memórias Inteiras, Memórias Reais e Contatos Auxiliares. Para cada uma destas opções existe uma janela de especificação dos identificadores dos operadores similar à encontrada no menu de Documentação. Para maiores detalhes sobre estas janelas de especificação, veja a descrição dos itens referentes à Documentação.

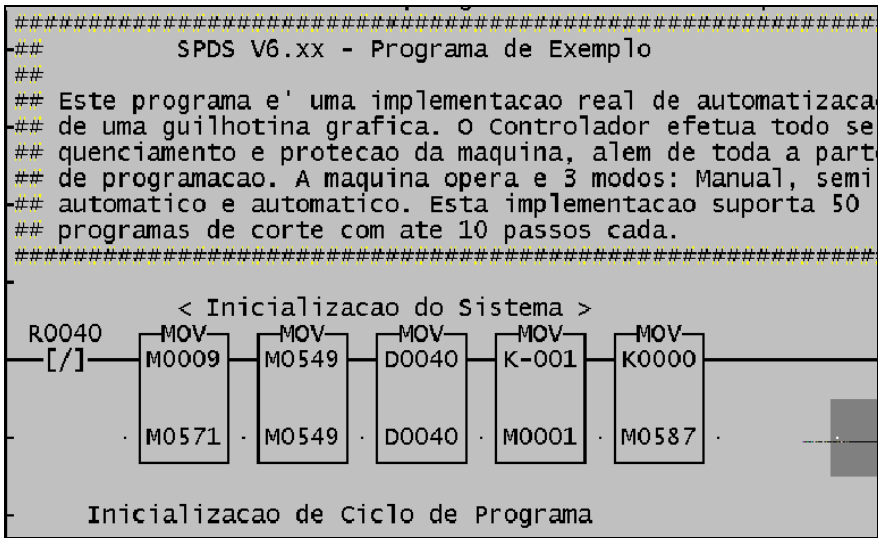
7.2.7 Documentação de uma Lógica - (F5)

Durante a confecção de um programa é importante para o usuário saber, durante o desenvolvimento de sua aplicação ou até muito tempo depois, o que uma determinada lógica significa e qual o seu propósito. O SPDS atende a este requisito oferecendo ao usuário o recurso de documentação de lógicas.

O editor de diagramas *Ladder* permite que o usuário introduza até três linhas, com 63 caracteres cada, de documentação/comentários por lógica. Têm-se acesso a este recurso através da tecla "F5-Document." do menu de comandos do editor.

Ao se teclar **F5**, aparece na tela um menu com as seguintes opções:

- Insere Documentação
- Edita Documentação
- Elimina Documentação
- Inibe Documentação



Insere Documentação

O usuário para documentar uma lógica qualquer deve inicialmente posicionar o cursor na linha inicial da lógica, teclar **F5** e escolher a opção "Insere Documentação" do menu que se abrir. Ao se selecionar esta opção, abrem-se três linhas de inserção de documentação/comentários na tela. O usuário pode inserir qualquer mensagem que desejar nestas três linhas.

Para auxiliar o usuário na edição de sua documentação, existem os seguintes recursos disponíveis, mostrados na linha de comandos:

- **ALT-K** - Elimina Linha

  Permite ao usuário eliminar a linha de documentação/comentário em que estiver o cursor.
- **ALT-R** - Repete caracter

  Permite repetir o último caracter teclado, quantas vezes desejar, bastando para isso manter estas teclas pressionadas.
- **INS** - Modo de Edição

  Permite ao usuário mudar o modo de edição, de Inserção para Sobrescrito e vice e versa.
- **ESC** - Fim

  Ao terminar a edição de sua documentação, ao se teclar ESC, o editor conclui a edição da documentação/comentários e retorna para o menu de documentação apresentado acima.

Edita Documentação

Após o usuário ter inserido alguma documentação no seu programa, é possível alterá-la. Isto é possível através da opção "Edita Documentação". Para isto é necessário que o usuário localize o cursor na linha da lógica em que existe a documentação a ser modificada, teclar **F5** e escolher a opção "Edita Documentação". A partir deste momento estarão válidos todos os recursos de edição como apresentados no item acima (Insere Documentação). Ao final da edição basta teclar-se ESC que o editor concluirá a edição e voltará para o menu de Edição da Documentação.

Elimina Documentação

Para que o usuário consiga eliminar uma determinada documentação é necessário que se coloque o cursor na primeira linha da lógica que se deseja eliminar o comentário e teclar **F5,** selecionando a opção "Eliminar Documentação". A documentação será automaticamente eliminada.

Inibe Documentação

O editor de diagramas Ladder permite ao usuário, a qualquer instante, inibir a apresentação das linhas de documentação das lógicas. Para isto é necessário que se tecle **F5**, selecionando-se a seguir a opção "Inibe Documentação". A partir deste momento o editor passará a mostrar apenas as lógicas do programa do usuário sem as linhas de documentação. O menu de Documentação passará a mostrar como última a seguinte opção:

- Ativa Documentação
  Permite ativar a apresentação da documentação das lógicas, uma vez que esteja inibidas.

É importante salientar que estes comandos de Inibição e Ativação da documentação não eliminam a documentação existente, elas apenas atuam na apresentação das mesmas.

7.2.8 Edição de operadores dos blocos - (F6)

É possível alterar os operadores de um bloco qualquer no programa do usuário. Ao se pressionar a tecla F6, permite-se ao usuário alterar o valor do operador indicado pelo cursor. Por exemplo, se o cursor estiver sobre um contato normalmente fechado, cujo identificador for I0020, ao se teclar **F6**, pode-se alterar este para I0032 ou até mesmo mudar o operador, se o tipo do bloco assim o permitir. No caso de se desejar modificar os operadores de um bloco, deve-se repetir o procedimento acima para cada operador do bloco que se desejar modificar. Este comando aparece no campo de comandos como "F6-Edita".

7.2.9 Sair do editor salvando o programa - (F8)

Para sair do editor salvando as modificações feitas no programa é necessário que o usuário pressione a tecla **F8**. Antes de sair do modo de edição, o programa é analisado sintaticamente e se estiver correto é automaticamente salvo em disco. No caso de se encontrar algum erro, o cursor é posicionado na ocorrência e uma mensagem explicativa sobre o erro é apresentada ao usuário. Este comando é mostrado na janela de auxílio do editor como "F8-Fim".

7.2.10 Inserção de uma linha em branco - (ALT A)

Este comando é útil quando se deseja inserir uma nova lógica entre duas lógicas consecutivas no programa e não existe espaço entre elas para a inserção. Este comando permite ao usuário abrir espaço entre as lógicas, para inserção de novos elementos, inserindo uma linha em branco (sem blocos) na posição do cursor. Neste caso, se houver alguma lógica abaixo da posição do cursor, ela será deslocada para baixo, abrindo um espaço entre a linha corrente (do cursor) e a próxima linha. A inserção é conseguida ao se pressionar simultaneamente as teclas **ALT** e **A**.

7.2.11 Eliminação de uma linha - (ALT E)

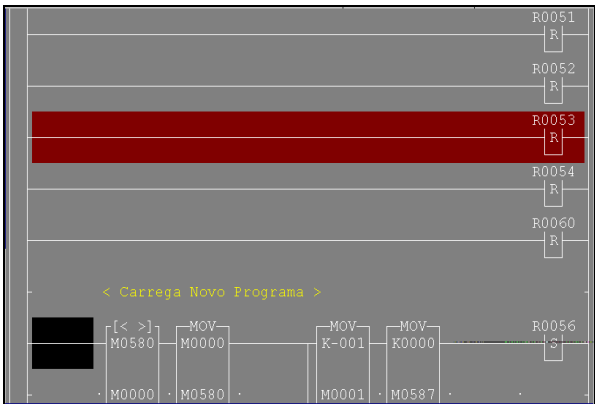
Durante a edição de um programa às vezes é necessário eliminar uma linha inteira no campo de edição. Este comando permite ao usuário eliminar a linha onde estiver localizado o cursor. Neste caso, se houver alguma lógica abaixo da posição do cursor, ela será deslocada uma linha para cima, perdendo-se o conteúdo da linha corrente. A eliminação é conseguida ao se pressionar simultaneamente as teclas **ALT** e **E.**

7.2.12 Próximo operador disponível - (ALT D)

Lista de Operadores Disponiveis
Tipo do Operador | Disp.
Contato Aux. Volatil | R4
Contato Aux. nao Volatil | R-1
Memoria Int. Volatil | M560
Memoria Int. nao Volatil | M6
Memoria Real Volatil | D15
Memoria Real nao Volatil | D10
Tabela de constante Real | Q0
Entrada Digital | I6
Saida Digital | O9
Entrada Analogica | E-1
Saida Analogica | S-1
ESC - Termina

Durante a edição de um programa às vezes se faz necessário saber qual o próximo operador disponível para utilização. Ao se pressionar simultaneamente as teclas **ALT** e **D**, abre-se uma janela na tela onde são mostrados todos os tipos de operadores e os seus respectivos índices correspondendo ao próximo disponível ao usuário para utilização no programa. Para os operadores M (memória) e R (contato auxiliar) são mostrados dois valores, correspondendo ao próximo elemento retentivo e ao não retentivo. Este recurso é muito útil, já que em programas muitos grandes é comum o usuário ficar sem saber qual o próximo operador que pode utilizar. O valor -1 associado a um campo indica que não existe operador daquele tipo disponível.

7.2.13 Cópia de áreas - (ALT I, F, L, C e R)



Um recurso que pode se tornar muito interessante para o operador é a cópia de áreas dentro do campo de edição. Caso o usuário tenha que editar várias lógicas muito parecidas umas com as outras, pode-se editar apenas uma, marcá-la como se fosse uma área e copiá-la para um *buffer* auxiliar e então posicionar o cursor onde se deseja uma cópia desta e comandar a cópia do *buffer* para a posição do cursor quantas vezes for necessário.

Para se marcar uma área na janela de edição é necessário indicar as posições inicial e final. A posição inicial é marcada colocando-se o cursor no local desejado e pressionando-se simultaneamente as teclas **ALT** e **I**, ao passo que para a final deve-se pressionar **ALT** e **F**. Ao se completar a marcação, a área demarcada será mostrada em vídeo reverso. Para copiar uma área para o *buffer* auxiliar, deve-se pressionar **ALT** e **L**. Para transferir o conteúdo do *buffer* auxiliar para a posição indicada pelo cursor, deve-se teclar **ALT** e **C**. Para desmarcar uma área tecla-se **ALT** e **R**.

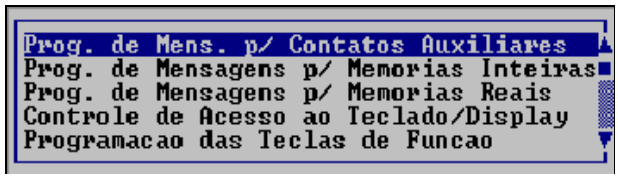
A cópia de áreas é realizada sempre em termos de linhas, não sendo possível copiar algo menos do que uma linha.

7.2.14 Especificação de mensagens da IHM - (ALT H)

Se o MCI02 que estiver sendo programado foi configurado para operar com a interface homem/máquina (módulo KBD240), pode-se especificar algumas mensagens que deverão ser mostradas no *display* em algumas situações. Ao se pressionar simultaneamente as teclas **ALT** e **H**, aparece um menu com as opções de mensagens que podem ser programadas.

A interface homem/máquina (IHM) formada pelo conjunto teclado/*display* permite ao usuário :

- Visualizar mensagem de identificação de sistema;
- Visualizar mensagens de sinalização de eventos;
- Supervisionar/Alterar as memórias existentes no programa do usuário;
- Supervisionar/Alterar o estado dos contatos auxiliares existentes no programa do usuário;
- (Des)Habilitar a programação de Memórias/Contatos Auxiliares através da IHM.


Prog. de Mens. p/ Contatos Auxiliares
Prog. de Mensagens p/ Memorias Inteiras
Prog. de Mensagens p/ Memorias Reais
Controle de Acesso ao Teclado/Display
Programacao das Teclas de Funcao

As memórias e contatos auxiliares podem ser supervisionados e/ou programados. Quando selecionado para supervisão, uma memória/contato auxiliar somente pode ter o seu valor mostrado no *display* do MCI02. Quando selecionado para programação, além da supervisão, o seu conteúdo pode ser alterado pelo usuário, através da interface homem/máquina do MCI02/MCI03/ZAP-500. Os tipos de mensagens e a maneira de programá-las são mostrados nos itens a seguir.

7.2.14.1 Mensagens de Programação de Contatos Auxiliares

Esta mensagem aparecerá no *display* da IHM do MCI sempre que se for supervisionar/alterar o estado de algum contato auxiliar definido no programa do usuário. Para exemplificar, podemos ter a seguinte mensagem exibida no *display* quando desejarmos programar/supervisionar um contato auxiliar :

AUTOM. / MAN.
[*] - *ON* [ ] - *OFF*

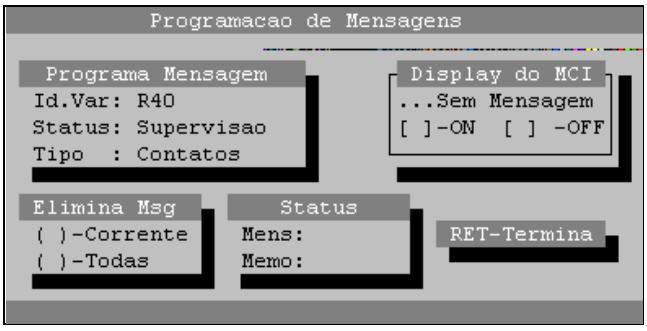
A mensagem que aparece na primeira linha do *display* é definida pelo usuário, ao passo que a da segunda linha é pré-definida pelo sistema, não sendo passível de alteração pelo usuário.

Se o usuário não definir nenhuma mensagem, será mostrado no *display* a seguinte mensagem:

...Sem Mensagem !
[*] - *ON* [ ] - *OFF*

Ao se selecionar a opção "Contatos Auxiliares", ou ao se posicionar o cursor sobre um contato auxiliar (operador R) e teclar-se **ALT** e **T**, abre-se uma janela na tela que permite ao usuário :

- Eliminar todas as mensagens;
- Eliminar a mensagem corrente;
- Inserir uma nova mensagem para ser associada a um contato auxiliar;
- Programar o estado de apresentação do contato auxiliar : Programação ou Supervisão;



Além disto, está disponível para o usuário nesta tela, o número total de mensagens já especificadas e o total de memória correspondente que será alocada no MCI02 para este fim.

7.2.14.2 Mensagem de Programação de Memórias Inteiras

Esta mensagem aparecerá no *display* da IHM do MCI sempre que se for supervisionar/alterar o conteúdo de alguma memória inteira definida no programa do usuário. Para exemplificar, podemos

ter a seguinte mensagem exibida no *display* quando desejarmos programar/supervisionar uma memória inteira :

TEMPO DE RESFR. Valor: xxxx

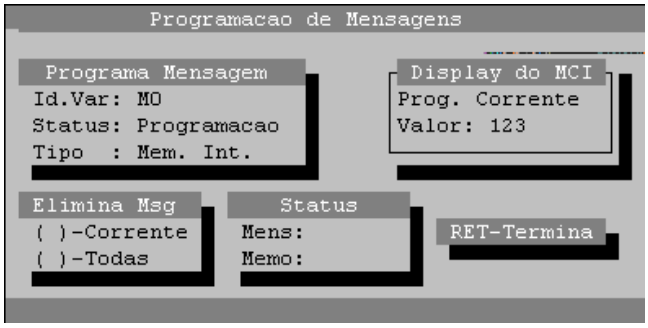
A mensagem que aparece na primeira linha do*display* é definida pelo usuário, ao passo que a da segunda linha é pré-definida pelo sistema, não sendo passível de alteração pelo usuário.

Se o usuário não definir nenhuma mensagem, será mostrado no*display* a seguinte mensagem:

...Sem Mensagem ! Valor: xxxx

Ao se selecionar a opção "Memórias Inteiras", ou ao se posicionar o cursor sobre uma memória inteira (operador M) e teclar-se **ALT** e **T**, abre-se uma janela na tela que permite ao usuário:

- Eliminar todas as mensagens;
- Eliminar a mensagem corrente;
- Inserir uma nova mensagem para ser associada a uma memória inteira;
- Programar o estado de apresentação da memória inteira : Programação ou Supervisão;



Além disto, está disponível para o usuário nesta tela, o número total de mensagens já especificadas e o total de memória correspondente que será alocada no MCI para este fim.

7.2.14.3 Mensagem de Programação de Memórias Reais

Esta mensagem aparecerá no*display* da IHM do MCI sempre que se for supervisionar/alterar o conteúdo de alguma memória real definida no programa do usuário. Para exemplificar, podemos ter a seguinte mensagem exibida no *display* quando desejarmos programar/supervisionar uma memória real :

TEMPERATURA Valor: xxx.xx

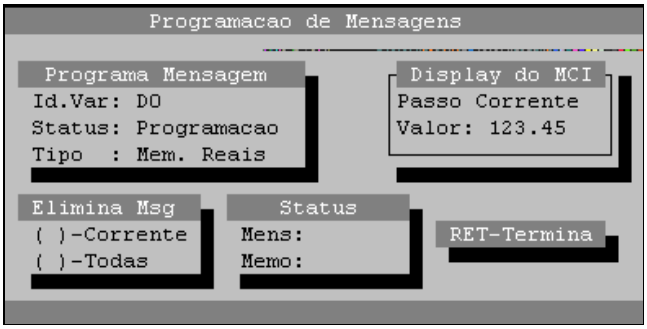
A mensagem que aparece na primeira linha do*display* é definida pelo usuário, ao passo que a da segunda linha é pré-definida pelo sistema, não sendo passível de alteração pelo usuário.

Se o usuário não definir nenhuma mensagem, será mostrado no*display* a seguinte mensagem:

...Sem Mensagem ! Valor: xxxx

Ao se selecionar a opção "Memórias Reais", ou ao se posicionar o cursor sobre uma memória real (operador D) e teclar-se **ALT** e **T**, abre-se uma janela na tela que permite ao usuário :

- Eliminar todas as mensagens;
- Eliminar a mensagem corrente;
- Inserir uma nova mensagem para ser associada a uma memória real;
- Programar o estado de apresentação da memória real : Programação ou Supervisão;



Além disto, está disponível para o usuário nesta tela, o número total de mensagens já especificadas e o total de memória correspondente que será alocada no MCI para este fim.

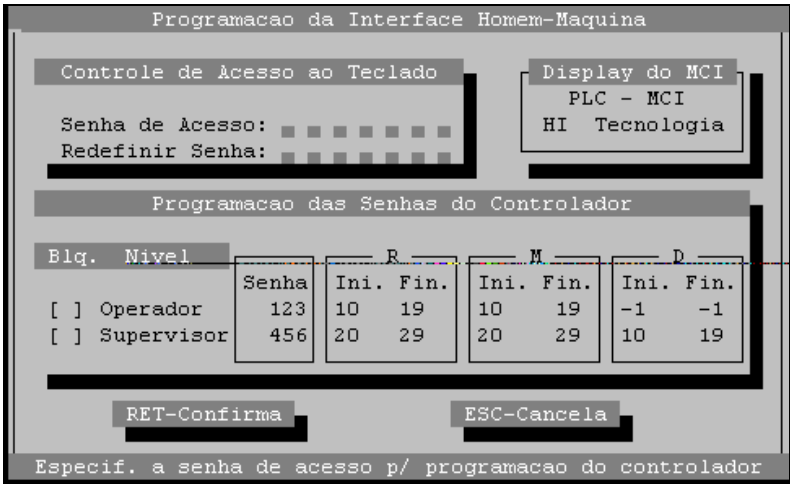
7.2.14.4 Mensagem de Sistema

Sempre que o MCI for energizado a mensagem mostrada abaixo, denominada "Mensagem de Sistema", aparecerá no *display* da IHM do MCI e deverá permanecer até que seja mostrada outra mensagem. Se o usuário desejar mostrar outra mensagem, pode-se especificá-la através da opção "Mensagem de Sistema".

PLC - MCI
H I Tecnologia

O acesso à interface homem/máquina do MCI pode ser controlado através do SPDS. Ao teclar-se **ALT H**, abre-se um menu com algumas opções para o usuário, entre elas está a opção "Controle de Acesso ou Teclado*Display*" que permite, através de uma janela de programação, o usuário inserir uma nova mensagem de sistema (campo*Display* do MCI). A mensagem de sistema é formada por duas linhas de dezesseis caracteres cada.

Nesta janela também é possível especificar o controle de acesso ao teclado e a programação de senhas do controlador, como está descrito nos dois itens a seguir.



7.2.15 Controle de Acesso ao Teclado

O acesso à interface homem/máquina do MCI pode ser controlado através do SPDS. Ao teclar-se **ALT H**, abre-se um menu com algumas opções para o usuário, entre elas está a opção

"Controle de Acesso ou Teclado*Display*" que permite, através de uma janela de programação, especificar o controle de acesso ao teclado (veja figura do item 7.2.14.4).

Para que o usuário tenha acesso à programação das senhas do controlador, é necessário que ele informe ao SPDS qual é a senha de acesso (campo "Senha de Acesso"). Esta senha é alfabética (somente pode conter letras) e deve ter no máximo 10 (dez) caracteres. Uma vez que o usuário forneceu a senha de acesso correto, ele pode passar a programar as senhas do controlador.

Uma senha de acesso já é fornecida junto com o programa SPDS e pode ser alterada pelo usuário através do campo "Redefinir Senha". Se o usuário redefinir a senha original fornecida, passa a ser sua responsabilidade guardar esta nova senha.

A senha originalmente fornecida pelo sistema é na verdade a ausência de senha, ou seja, o usuário pode ter acesso à programação das senhas do controlador sem fornecer qualquer senha. A partir do momento que ele redefinir esta senha é que o sistema restringirá o acesso à programação das senhas do controlador.

A programação de senhas do controlador será explicada no próximo item.

7.2.16 Programação de Senhas do Controlador

A programação de senhas do controlador pode ser feita através do comando **ALT H** do editor, selecionando-se a seguir a opção "Controle de Acesso ou Teclado*Display*", como explicado no item anterior (veja figura do item 7.2.14.4).

Pode-se especificar senhas de acesso para memórias reais, inteiras e contatos auxiliares. Esta senha é especificada para uma determinada faixa de memória, ou contatos auxiliares, por exemplo, pode-se especificar que as memórias de M0 a M10 poderão ter acesso via IHM do MCI02 somente se a senha especificada for habilitada. Caso contrário não se poderá ter acesso a estas memórias.

As senhas de acesso são especificadas em dois níveis: do operador e do supervisor. O nível do supervisor tem maior prioridade do que a do operador, ou seja, se uma memória qualquer estiver especificada para ter uma senha para o operador e uma para o supervisor, a senha e os comandos de habilitação/bloqueio do supervisor prevalecerão sobre a do operador. As senhas do operador e supervisor devem ser numéricas e podem ter no máximo 5 (cinco) caracteres.

Para cada contato auxiliar (R), memória inteira (M) e memória real (D) pode ser especificada uma faixa de controle de acesso, para o operador e para o supervisor. Esta faixa é especificada fornecendo-se os seus índices inicial e final.

A tabela mostrada a seguir fornece uma visão geral do processo de acesso às variáveis através da IHM do MCI quando as senhas de acesso para o operador e supervisor estão habilitadas e bloqueadas.

Situação da Senha	Acesso à Programação	Acesso à Supervisão
Supervisor Habilitado	Programação permitida	Supervisão permitida
Supervisor Bloqueado	não mostra nada	não mostra nada
Operador Habilitado	Programação permitida	Supervisão permitida
Operador Bloqueado	Supervisão permitida	Supervisão permitida

Para que um operador ou supervisor tenha acesso as variáveis especificadas, deve-se agir da seguinte maneira:

Supervisor

A primeira providência que o supervisor deve tomar para ter acesso às variáveis que possuam senha é habilitar a mesma no MCI.

Isto é conseguido teclando-se **F** (função) no teclado do MCI02 e selecionando-se, através da tecla **M** (menu) a opção "Senha de Acesso". Ao ser selecionada esta opção, o MCI pede ao usuário que especifique se a senha é do Operador ou do Supervisor, que pode ser escolhida teclando-se a tecla M (menu).

Uma vez selecionada a opção "Supervisor", através da tecla **E** (entra), é pedido ao usuário que especifique se deseja Habilitar ou Bloquear o acesso às variáveis. A seguir é pedido ao usuário que forneça a senha do supervisor. Uma vez que for fornecida pelo usuário e estando correta, o MCI avisará que a mesma está habilitada e permitirá a partir deste instante que as variáveis especificadas sejam acessadas sem restrição. O acesso às variáveis pode ser: programação ou supervisão, como descrito no item 7.2.14 “Especificação de mensagens da IHM (ALTH)”.

Para que o supervisor bloqueie o acesso às variáveis, deve-se repetir o processo descrito acima, só que deve-se selecionar a opção de "bloquear" o acesso.

Operador

A primeira providência que o operador deve tomar para ter acesso às variáveis que possuam senha de acesso é habilitar a mesma no MCI.

Isto é conseguido teclando-se **F** (função) no teclado do MCI e selecionando-se, através da tecla **M** (menu) a opção "Senha de Acesso". Ao ser selecionada esta opção, o MCI pede ao usuário que especifique se a senha é do Operador ou do Supervisor, que pode ser escolhida teclando-se a tecla M (menu).

Uma vez selecionada a opção "Operador", através da tecla **E** (entra), é pedido ao usuário que especifique se deseja Habilitar ou Bloquear o acesso às variáveis. A seguir é pedido ao usuário que forneça a senha do operador. Uma vez que for fornecida pelo usuário e estando correta, o MCI avisará que a mesma está habilitada e permitirá a partir deste instante que as variáveis especificadas sejam acessadas sem restrição. O acesso às variáveis pode ser: programação ou supervisão, como descrito no item 7.2.14 “Especificação de mensagens da IHM (ALTH)”.

Para que o operador bloqueie o acesso às variáveis, deve-se repetir o processo descrito acima, só que deve-se selecionar a opção de "bloquear" o acesso.

7.2.17 Programação de Mensagens de Eventos (ALT T)

O SPDS permite ao usuário sinalizar a ocorrência de eventos através da interface homem/máquina do MCI. Esta sinalização é comandada através do programa do usuário. Existe um bloco sinalizador de eventos que pode ser inserido no lugar de uma bobina (oitava coluna do editor de diagramas *Ladder*). Uma vez que este bloco estiver inserido no programa, pode-se especificar uma mensagem de sinalização associada ao evento.

O sistema prevê uma prioridade para cada mensagem de alarme. Uma mensagem menos prioritária deve ser sobreposta por uma mais prioritária. A mensagem de número 0 (zero) é a mais prioritária.

Uma mensagem aparecerá no *display* sempre que a entrada do bloco de sinalização de evento correspondente estiver energizada e deixará de ser mostrada quando ficar desenergizada. Para exemplificar, podemos ter a seguinte mensagem mostrada no *display* quando da ocorrência de um evento qualquer :

A L A R M E TEMP. ALTA

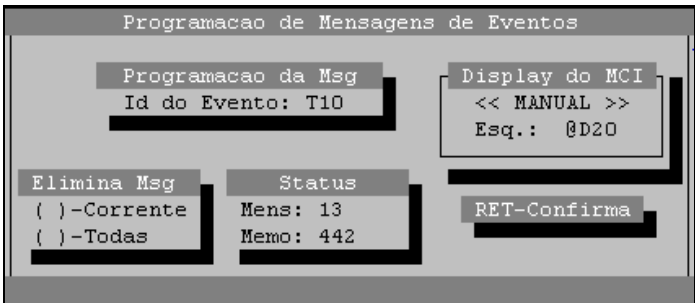
Obs: a mensagem mostrada acima foi especificada pelo usuário.

Especificação de Mensagens de Sinalização de Eventos

Para especificar uma mensagem de sinalização de eventos, deve-se localizar o cursor do editor sobre o bloco de sinalização de eventos e teclar simultaneamente as teclas **ALT** e **T**. A partir deste instante é aberta uma janela de programação de mensagens de eventos na tela do editor.

Na janela de especificação de mensagens de eventos existem vários campos que além de permitir a especificação da mensagem de sinalização do evento (campo *Display* do MCI), também mantêm o usuário informado sobre a quantidade de mensagens existentes e a quantidade de memória utilizada por elas (campo de Status). Como podem existir várias mensagens associadas à eventos, deve-se especificar qual a mensagem que se desejar inserir. Isto é feito através do campo "Programação da Msg", fornecendo o identificador do Evento (operador Ax, onde x corresponde ao número do bloco de sinalização de eventos que a mensagem estiver associada).

Pode-se eliminar a mensagem corrente ou todas as mensagens associadas aos blocos sinalizadores de eventos (campo Elimina Msg). Para especificar a mensagem, basta posicionar o cursor sobre o campo que simula o *display* do MCI (campo *Display* do MCI) e digitar a mensagem, que pode ter duas linhas de dezesseis caracteres cada.



Adicionalmente, para as mensagens de sinalização de eventos, temos a possibilidade de mostrar o conteúdo de uma memória (inteira ou real) ou de um contato auxiliar na segunda linha do *display* do MCI02. Para que isto ocorra, é necessário que se especifique a memória ou contato auxiliar que terá o seu conteúdo mostrado.

O sistema permite que se especifique os operadores M, D e R. Estes operadores devem ser especificados na segunda linha da mensagem de sinalização de eventos antecedidos do caracter especial @, que sinaliza esta situação. Após o caracter @, o sistema substituirá o operador pelo seu conteúdo durante a sinalização do evento. Na figuras mostradas a seguir, temos um exemplo de mensagem quando de sua especificação e durante a sinalização da ocorrência de um evento.

A L A R M E TEMP (%): @D5

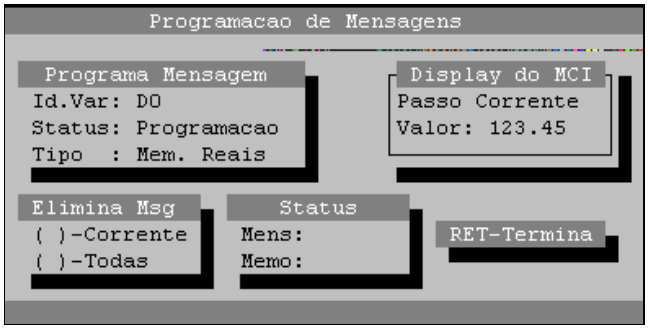
A mensagem acima especificada, será mostrada durante a ocorrência do evento, supondo-se que o valor de D5 é 82,50, da seguinte maneira :

A L A R M E TEMP (%): 82,50

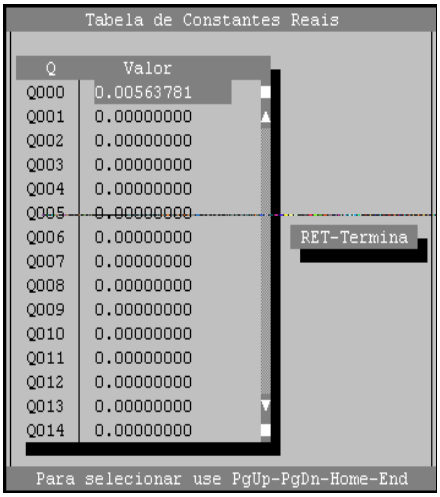
Para a especificação destes operadores fazemos as seguintes considerações :

- A especificação deve ser feita apenas na segunda linha da mensagem, correspondendo à segunda linha do *display* do MCI;
- O espaço reservado para a exibição do valor é considerado a partir do caracter @ até o final da linha e o usuário é responsável pela sua especificação correta, já que ele deve saber com quantos dígitos é possível mostrar o valor requisitado;
- Só é permitido especificar um único operador por mensagens de alarme;
- Após o caracter @ somente é permitido se especificar um operador M, D ou R;
- Os contatos auxiliares têm o seu estado mostrado como : [*ON*] ou [*OFF*];

Uma vez especificadas as mensagens, ao teclar-se **RETURN**, a janela se fechará e as mensagens passarão a fazer parte do programa do usuário.



7.2.18 Especificação de Constantes Reais



Quando se utiliza uma constante real como operador em um bloco, dentro do editor de diagramas *Ladder*, ela é especificada por um índice que pode variar de 0 a 99, como por exemplo Q01 ou Q12. Os valores correspondentes destas constantes reais são introduzidos através de comando **ALT Q**.

Ao se digitar **ALT Q**, abre-se uma janela no vídeo, na qual existe uma coluna que contêm as constantes de Q00 a Q99 e uma outra coluna, em frente, reservada para se introduzir o valor correspondente da constante.

7.2.19 Alteração da base numérica (ALT B)

Durante a edição de um programa é interessante e as vezes necessário poder especificar valores na base decimal e ou na hexadecimal. A base corrente é mostrada na linha superior do editor de diagramas *Ladder*.

Para alternar entre a base decimal e hexadecimal, deve-se pressionar simultaneamente as teclas **ALT** e **B**. Se a base corrente é decimal, ao teclar-se **ALT B**, ela passará para hexadecimal e vice e versa. Uma vez escolhida a base, esta permanecerá até o fim da edição do programa ou até que o usuário resolva alterá-la.

7.2.20 Troca de um contato [NA por NF] ou [NF por NA] (ALT N)

Durante a edição de um programa às vezes o usuário gostaria que no lugar de um contato NA em sua lógica existisse um contato NF, ou vice e versa. Para tanto, o usuário precisa teclar simultaneamente as teclas **ALT** e **N**, que esta alteração se dará. O contato NA que estiver sob o cursor se transformará em um contato NF e um contato NF se transformará em um NA.

7.2.21 Salvar programa em disco - (ALT S)

Para salvar o programa que se estiver editando, basta pressionar simultaneamente as teclas **ALT** e **S** que o editor procederá a uma análise do programa editado e se ele não contiver erros, o programa será salvo em disco. Caso contrário a situação que gerou o erro será informada ao usuário e o cursor será posicionado onde o erro foi detectado.

Antes de salvar o programa, é pedido ao usuário o nome do projeto no qual se deseja salvar o programa. Desta forma, oferece-se ao usuário a oportunidade de se salvar o programa em um outro projeto diferente do qual se está correntemente trabalhando.

7.2.22 Obter informações sobre elemento/bloco (ALT F1)

Permite ao usuário obter informações sobre um bloco/elemento específico. Para isto deve-se posicionar o cursor sobre o elemento/bloco que desejar e teclar simultaneamente as teclas **ALT** e **F1**. A partir deste instante é aberta uma janela na tela com todas as informações pertinentes ao elemento/bloco.

7.2.23 Sair do editor sem salvar o programa (ESC)

Pode-se sair do editor de programas sem salvar qualquer modificação efetuada no programa pelo usuário, bastando para isso teclar-se **ESC**. Para evitar qualquer acidente durante a digitação, para que este comando seja executado é pedido ao usuário que confirme a sua decisão.

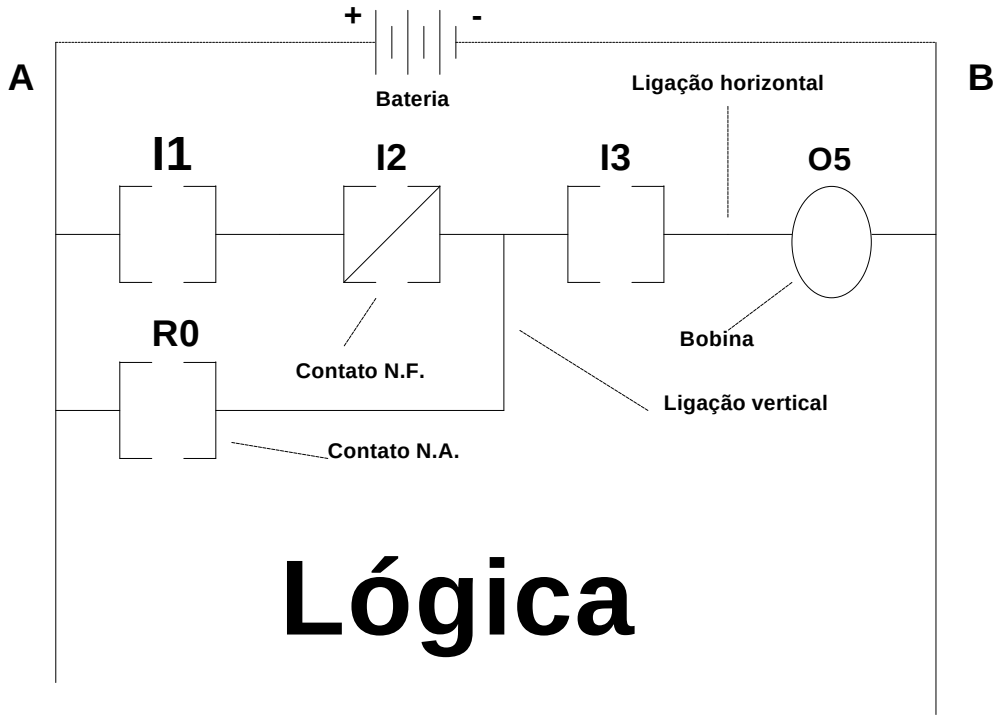
8. Diagramas *Ladder*

Um diagrama *Ladder* ou diagrama de relés é uma representação gráfica de ligações lógicas feita através de ligações de relés. Este tipo de diagrama representa a ligação entre relés necessária para que uma determinada função seja executada. Este diagrama é normalmente utilizado em esquemas de painéis de controle e intertravamento de sinais de máquinas e processos e é constituído basicamente pela interligação de relés de vários tipos.

Em um diagrama *Ladder* uma lógica pode ser constituída por vários blocos. Estes blocos podem ser: contatos normalmente abertos (**-[ ]-**) e normalmente fechados (**-[/]-**) de um relé, sua respectiva bobina (**-( )-**), conexões horizontais (**----**), verticais (**|**) e blocos funcionais.

Um relé, que é um elemento fundamental na elaboração de um diagrama de relés, neste sistema é implementado podendo ter vários contatos, que podem ser normalmente abertos ou normalmente fechados. Na figura abaixo, temos uma representação de um diagrama de relés. Nele podemos identificar vários elementos, tais como :

- Bateria para energização do circuito formado pelos relés.
- Contatos normalmente abertos de alguns relés (R0, I1 e I3).
- Contatos normalmente fechados de um relé (I2).
- Bobina de um relé (O5).



Conceitos Básicos de um Relé

Antes de começarmos a explicar o funcionamento do circuito, temos que apresentar alguns aspectos básicos do funcionamento de um relé.

- Um relé é formado por uma bobina e por contatos, que podem ser normalmente fechados ou normalmente abertos.
- Um contato de um relé funciona como uma chave em um circuito elétrico, se estiver aberta o circuito será interrompido. Se estiver fechado, o circuito elétrico terá continuidade.
- Uma bobina de um relé desenergizada faz com que os seus contatos normalmente abertos estejam abertos (estado original) e os contatos normalmente fechados estejam fechados (estado original).
- Uma bobina de um relé energizada faz com que os seus contatos normalmente abertos se fechem e os contatos normalmente fechados se abram.
- O número de contatos normalmente abertos ou fechados de um relé é definido durante a construção do mesmo.

Funcionamento da Lógica de Relés

No diagrama apresentado na figura anterior, temos uma bobina (O5) e uma lógica cujo objetivo final é decidir, através do circuito formado pelos contados dos relés R0, I1, I2 e I3, se a bobina será energizada ou não.

Eletricamente a bobina será energizada se ele estiver com os seus dois terminais ligados à bateria. Isto somente será possível se a combinação lógica formada pelos contatos dos relés no circuito assim o permitir.

A bobina será energizada na seguinte condição:

Se ((I1 estiver fechado e I2 estiver aberto) ou R0 estiver fechado) e I3 estiver fechado

Para que esta condição aconteça, devemos ter uma das duas possibilidades apresentadas abaixo ou as duas simultaneamente :

- O contato normalmente aberto do relé I1 deve estar fechado (sua bobina deve estar energizada) e o contato normalmente fechado do relé I2 deve estar em seu estado original, ou seja, fechado (sua bobina dever estar desenergizada) e ainda o contato normalmente aberto do relé I3 deve estar fechado (sua bobina deve estar energizada).

OU

- O contato normalmente aberto do relé R0 deve estar fechado (sua bobina deve estar energizada) e o contato normalmente aberto do relé I3 deve estar fechado (sua bobina deve estar energizada)

Desta forma temos uma lógica feita por contatos de relés ditando o estado de outros contatos de um outro relé, através da energização de sua bobina. Assim, podemos construir desde uma lógica simples até as mais complexas em um diagrama de relés.

Os contatos I1, I2 I3 e a bobina O5 estão ligados em série e o contato R0 está ligado em série com o contato I3 e a bobina O5 e em paralelo com os contatos I1 e I2.

Diagrama de contatos de relés no SPDS

Como se trata de uma versão mais moderna e aprimorada com os recursos da computação, neste diagrama existem, memórias, valores analógicos de entrada ou saída, blocos funcionais implementando várias funções complexas, além de contatos e bobinas de relés.

Em termos de relé, podemos ter três tipos de negrito:

- **R** - relé auxiliar, ficando o seu estado associado à uma memória do computador (MCI). Sempre que se deseja colocar em uma lógica o estado de um relé que não esteja associado à uma entrada ou saída digital, ela será representada por um contato de um relé R.

- **I** - relé associado a uma entrada digital do computador (MCI). Sempre que se deseja colocar em uma lógica o estado de uma entrada digital, ela será representada por um contato de um relé I.

- **O** - relé associado a uma saída digital do computador (MCI). Sempre que se deseja colocar em uma lógica o estado de uma saída digital, ela será representada por um contato de um relé O.

No diagrama, um contato de um relé e a sua bobina possuem a mesma identificação, ou seja, a bobina de um relé R0010 é referenciada da mesma maneira que um de seus contatos. Esta regra vale para os três tipos de relés apresentados acima.

Em um diagrama de relés, dependendo do tipo da bobina do relé e da aplicação, a fonte de alimentação pode não ser uma bateria, mas uma fonte alternada de corrente.

No editor de diagramas *Ladder* do SPDS a bateria usada para a energização do circuito não é representada, aparecendo apenas as linhas **A** e **B** mostradas no circuito, ficando portanto subentendida a sua existência. Em uma linha de edição podem ser colocados em série no máximo até sete contatos de relés e uma bobina.

Através de blocos são implementadas algumas funções tais como : Contadores, Temporizadores, Somadores, blocos de controle PID e etc. A seguir temos uma descrição de todos os blocos disponíveis no editor e as regras de inserção do elemento na lógica.

- **Conexões**
  - **Conexão Vertical**
  - **Conexão Horizontal**

- **Contatos**
  - **Contato Normalmente Aberto (NA)**
  - **Contato Normalmente Fechado (NF)**

- **Bobinas**
  - **Bobina Simples**
  - **Bobina SET**
  - **Bobina RESET**
  - **Bobina Pulso (Mono Estável)**
  - **Bobina Oscilador**

- **Movimentação de Dados**
  - **Movimentação Indexada**
  - **Movimentação Indexada**
  - **Inicialização de Dados**

- **Temporizador**

- **Contadores**
  - **Contador Unidirecional**
  - **Contador Bidirecional**

- **Funções Matemáticas**
  - **Soma**
  - **Subtração**
  - **Multiplicação**
  - **Divisão**
  - **Raiz Quadrada**
  - **Deslocamento à Esquerda**
  - **Deslocamento à Direita**

- **Operações Lógicas**
  - **Operador AND**
  - **Operador OR**
  - **Operador XOR**

- **Comparadores**
  - **Igual a**
  - **Diferente**
  - **Menor que**
  - **Menor ou igual a**
  - **Maior que**
  - **Maior ou igual a**
  - **Teste lógico AND**

- **Blocos de Controle de Programa**
  - **Fim de Programa**
  - **Inicio de Relé Mestre**
  - **Fim de Relé Mestre**
  - **Bloco de Lógica**
  - **Início de Bloco de Lógica**
  - **Fim de Bloco de Lógica**
  - **Bobina de Salto**

- **Interface Homem Máquina**
  - **Mensagem de Evento**

- **Especiais**
  - **Relógio de Tempo Real**
  - **Controle PID**
  - **Controle de Motor de Passo**
  - **Controle de Servo Motor**
  - **Interface de Encoder**
  - **Interface de Resolver**
  - **Gerador de Frequência**
  - **Contador de Alta Velocidade**
  - **Comunicação com PLC Remoto**
  - **Configuração/Status Controlador**

A seguir apresenta-se uma descrição de cada um destes elementos ou blocos de controle disponíveis para a especificação do programa ladder da aplicação.

8.1 Conexões

8.1.1 Conexão Horizontal

Objetivo - Este elemento tem como função fazer a interligação horizontal (em série) de elementos em uma lógica (**----**). Ele serve como um fio de ligação entre os relés e blocos funcionais.

Modo de Inserção do Elemento no programa

Posicionando-se o cursor na posição desejada e teclando-se a letra H

Símbolo do Elemento

Parâmetro

- Não tem

Descrição - Este elemento representa uma conexão horizontal no diagrama de relés, sendo um elemento fundamental na interligação dos elementos em uma lógica, completando o circuito elétrico da lógica.

Uma lógica normalmente utiliza este elemento, já que é sempre necessário completar o circuito elétrico entre os pontos A e B (veja item 8-Diagramas*Ladder*).

8.1.2 Conexão Vertical

Objetivo - Este elemento tem como função fazer a interligação vertical (em paralelo) de elementos em uma lógica (**|**). Ele serve como um fio de ligação entre os relés e blocos funcionais.

Modo de Inserção do Elemento no programa

Posicionando-se o cursor na posição desejada e teclando-se a letra V

Símbolo do Elemento

Parâmetros

- Não tem.

Descrição - Este elemento representa uma conexão vertical no diagrama de relés, sendo um elemento fundamental na interligação dos elementos em uma lógica, completando o circuito elétrico da lógica.

Uma lógica normalmente utiliza este elemento, já que é sempre necessário completar o circuito elétrico entre os pontos A e B (veja item 8-Diagramas*Ladder*).

8.2 Contatos

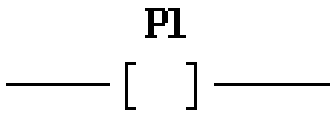
8.2.1 Contato Normalmente Aberto (NA)

Objetivo - O objetivo deste elemento é funcionar como um contato normalmente aberto de um relé. Se a bobina do relé estiver desenergizada, ele estará aberto interrompendo o circuito lógico. Se a sua bobina estiver energizada, ele estará fechado, deixando que o circuito se complete através dele.

Modo de Inserção do Elemento no programa

Posicionando-se o cursor na posição desejada e teclando-se a letra A

Símbolo do Elemento



Parâmetros

- **P1** - identifica o contato normalmente aberto como sendo de um relé do tipo I, R ou O.

Descrição - Este elemento pode ser colocado em qualquer posição dentro de uma lógica, exceto na oitava coluna da janela de edição que é reservada para as bobinas dos relés. Ele representa um contato normalmente aberto de um relé (**-[ ]-**).

Os operadores que este elemento aceita são : I, R e O. Se um contato NA estiver associado à um operador I, ele estará representado o estado de uma entrada digital. Se estiver associado à um operador R, ele estará representando o estado de um contato de um relé auxiliar e se estiver associado à um operador O, ele estará representando o estado de uma saída digital do MCI.

Observação: Como no MCI02 os módulos de entrada digital possuem 16 (dezesseis) canais, e o operador I representa um destes canais, os canais ou operadores numerados de 0 a 15 corresponderão ao primeiro módulo de entradas digitais (DIM), os numerados de 16 a 31, ao segundo módulo e assim por diante.

Observação: O MCI03 possui dois módulos de interface com o processo. Esses módulos são: MIO321 e MIO330. O módulo MIO321 possui oito entradas digitais, ao passo que o MIO330 possui dezenove entradas digitais. Os operadores I que representam estas entradas são os numerados de 0 a 7, para o caso do módulo MIO321 e os numerados de 0 a 18 corresponderão às entradas do módulo MIO330.

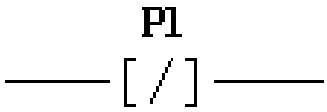
8.2.2 Contato Normalmente Fechado (NF)

Objetivo - O objetivo deste elemento é funcionar como um contato normalmente fechado de um relé. Se a bobina do relé estiver desenergizada, ele estará fechado, deixando que o circuito se complete através dele. Se a sua bobina estiver energizada, ele estará aberto, interrompendo o circuito.

Modo de Inserção do Elemento no programa

Posicionando-se o cursor na posição desejada e teclando-se a letra F

Símbolo do Elemento



Parâmetros

- **P1** - identifica o contato normalmente fechado como sendo de um relé do tipo I, R ou O.

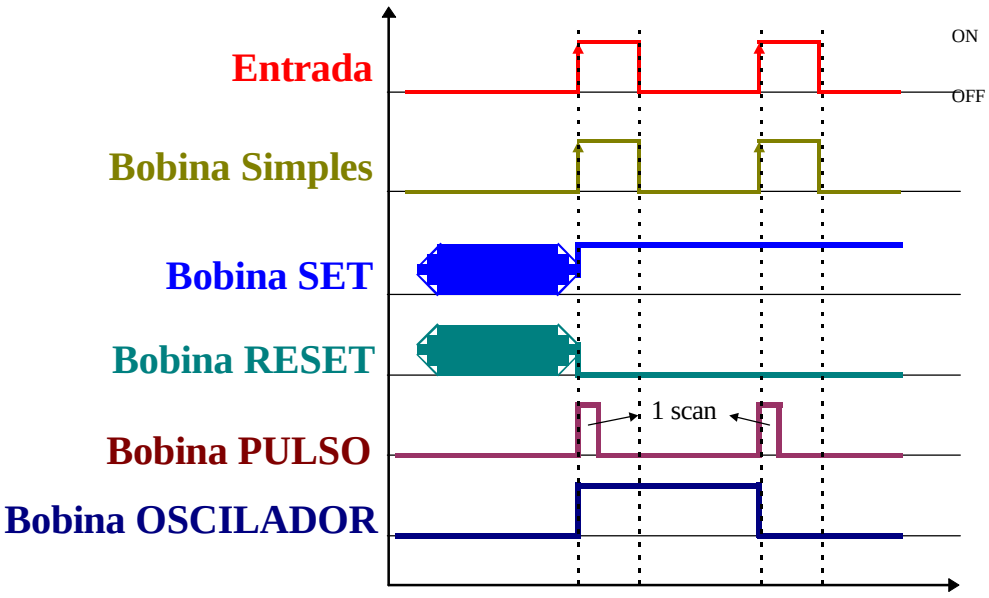
Descrição - Este elemento pode ser colocado em qualquer posição dentro de uma lógica, exceto na oitava coluna da janela de edição que é reservada para as bobinas. Ele representa um contato normalmente fechado de um relé (**-[/]-**). Os operadores que este elemento aceita são : I, R e O. Se um contato NF estiver associado à um operador I, ele estará representado o estado de uma entrada digital. Se estiver associado à um operador R, ele estará representando o estado de um contato de um relé auxiliar e se estiver associado à um operador O, ele estará representando o estado de uma saída digital do MCI.

Observação: Como mo MCI02 os módulos de entrada digital possuem 16 (dezesseis) canais, e o operador I representa um destes canais, os canais ou operadores numerados de 0 a 15 corresponderão ao primeiro módulo de entradas digitais (DIM), os numerados de 16 a 31, ao segundo módulo e assim por diante.

Observação: O MCI03 possui dois módulos de interface com o processo. Esses módulos são: MIO321 e MIO330. O módulo MIO321 possui oito entradas digitais, ao passo que o MIO330 possui dezenove entradas digitais. Os operadores I que representam estas entradas são os numerados de 0 a 7, para o caso do módulo MIO321 e os numerados de 0 a 18 corresponderão às entradas do módulo MIO330.

8.3 Bobinas

No caso dos equipamentos da linha HI tem-se disponíveis as seguintes bobinas : Bobina Simples, Bobina Set, Bobina Reset, Bobina de Pulso e Bobina Oscilador. O gráfico abaixo illustra as diferenças entre estes respectivos tipos de bobinas.



Considerando como entrada para as bobinas um sinal com duas transições, por exemplo, decorrentes de dois acionamentos de um botão de contato momentâneo (push-button). A bobina simples reproduz exatamente uma cópia do sinal de entrada, ou seja, quando a entrada esta ativa a bobina também está ativa e vice-versa.

A bobina set, independente do estado anterior, ao detectar a primeira transição de subida do sinal de entrada passa a saída da bobina para o estado ativo, permanecendo neste estado, mesmo a entrada passando para inativa. Neste caso, para desativar esta bobina é necessário, por exemplo, utilizar uma bobina reset.

A bobina reset opera de modo inverso a bobina set, ou seja, independente do estado anterior, ao detectar a primeira transição de subida do sinal de entrada passa a saída da bobina para o estado inativo, permanecendo neste estado, mesmo a entrada passando para inativa.

A bobina de pulso gera a cada transição de subida do sinal de entrada um pulso na respectiva bobina. Note-se que este pulso somente é detectado em um único “ciclo de scan” do programa ladder, ou seja, uma vez detectada a transição de subida da entrada, a partir deste ponto do programa ladder a bobina passa para ativa, e no próximo ciclo de scan do programa ladder ao passar por esta lógica novamente a bobina para o estado inativa.

A bobina oscilador opera de tal modo que a cada transição de subida do sinal de entrada inverte o seu respectivo estado. Assim, ao detectar uma transição de subida da entrada, se estava ativa passa para inativa, e se estava ativa passa para inativa. Ao mudar de estado a bobina permanece neste estado, mesmo a entrada passando para inativa, e somente inverte o estado corrente quando a entrada gerar uma nova transição de subida, e assim sucessivamente.

A seguir apresenta-se uma descrição mais detalhada de cada uma destas bobinas.

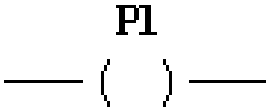
8.3.1 Bobina Simples

Objetivo - O objetivo deste elemento é funcionar como uma bobina de um relé. Se estiver energizada, todos os seus contatos normalmente abertos estarão fechados e os normalmente fechados estarão abertos. Se estiver desenergizada, todos os seus contatos normalmente abertos estarão abertos e os normalmente fechados estarão fechados. Não existe um limite de contatos que um relé pode ter, sendo NA ou NF. Para cada relé somente pode existir uma única bobina.

Modo de Inserção do Elemento no programa

Posicionando-se o cursor na posição desejada e teclando-se a letra B. Este elemento somente pode ser posicionado na oitava coluna do editor de programas.

Símbolo do Elemento



Parâmetros

- **P1** - identifica a bobina do relé, podendo ser a bobina de um relé do tipo R ou O.

Descrição - Este elemento representa uma bobina de relé **-( )-**). Só pode ser colocado na oitava coluna da janela de edição. Os operadores que este elemento aceita são : O e R. Se uma bobina estiver associada à um operador O, ele estará representando o estado de uma bobina de relé associada à uma saída digital do MCI. Se estiver associado à um operador R, ele estará representando o estado de uma bobina de um relé auxiliar.

Observação 1: Como no MCI02 os módulos de saída digital possuem 16 (dezesseis) canais, e o operador O representa um destes canais, os canais ou operadores numerados de 0 a 15 corresponderão ao primeiro módulo de saídas digitais (DOM ou RDM), os numerados de 16 a 31, ao segundo módulo e assim por diante.

Observação 2: O MCI03 possui dois módulos de interface com o processo. Esses módulos são: MIO321 e MIO330. O módulo MIO321 possui oito saídas digitais, ao passo que o MIO330 possui doze saídas digitais. Os operadores O que representam estas saídas são os numerados de 0 a 7, para o caso do módulo MIO321 e os numerados de 0 a 11 corresponderão às saídas do módulo MIO330.

8.3.2 Bobina *SET*

Objetivo - O objetivo deste elemento é operar como uma entrada*SET* de um "*Flip-Flop*", que é um elemento básico de memória em circuitos elétricos.

Modo de Inserção dos Elementos no programa

Posicionando-se o cursor na oitava coluna da linha desejada e teclando-se a letra G. Neste instante aparecerá um menu com sete opções, entre elas, temos “Bobina”. O usuário deve selecionar a esta opção e teclar “Enter”. Este elemento necessita de no mínimo uma linha livre para a sua inserção.

Símbolo do Elemento

Parâmetros

- **P1** - identifica o relé que terá a sua bobina energizada. Este relé pode ser em memória (operador R) ou uma saída digital (operador O).

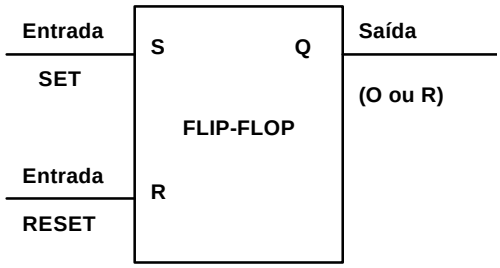
Entradas

- **E1** - entrada de habilitação (Habilita) do bloco de bobina *SET*. Sempre que esta entrada estiver energizada a bobina do relé especificado será energizada.

Saídas

- não tem.

Descrição - Este elemento está associado ao controle de saídas digitais (operadores Oxxxx) e contatos auxiliares (operadores Rxxxx) e só pode ser colocado na oitava coluna da janela de edição. Nesta situação a saída digital (ou contato auxiliar) está associada à saída de um "*FLIP-FLOP*", como mostrado na figura a seguir.



O elemento bobina *SET* atua na entrada *SET* do "*FLIP-FLOP*" da seguinte maneira :

- Caso o elemento esteja energizado, o operador associado (Oxxxx ou Rxxxx) será ativado.
- Se o elemento estiver desenergizado, o operador associado (Oxxxx ou Rxxxx) mantém o valor corrente inalterado.

Este elemento é utilizado em conjunto com o elemento Bobina *RESET*, descrito a seguir.

8.3.3 Bobina *RESET*

Objetivo - O objetivo deste elemento é operar como uma entrada *RESET* de um "*Flip-Flop*", que é um elemento básico de memória em circuitos elétricos.

Modo de Inserção dos Elementos no programa

Posicionando-se o cursor na oitava coluna da linha desejada e teclando-se a letra G. Neste instante aparecerá um menu com sete opções, entre elas, temos “Bobina”. O usuário deve selecionar a esta opção e teclar “Enter”. Este elemento necessita de no mínimo uma linha livre para a sua inserção.

Símbolo do Elemento

Parâmetros

- **P1** - identifica o relé que terá a sua bobina desenergizada. Este relé pode ser em memória (operador R) ou uma saída digital (operador O).

Entradas

- **E1** - entrada de habilitação (Habilita) do bloco de bobina (*RESET*). Sempre que esta entrada estiver energizada a bobina do relé especificado será desenergizada.

Saídas

- não tem.

Descrição - Este elemento está associado ao controle de saídas digitais (operadores Oxxxx) e contatos auxiliares (operadores Rxxxx). Nesta situação a saída digital (ou contato auxiliar) está associada à saída de um "*FLIP-FLOP*", como mostrado na figura anterior.

O elemento bobina *RESET* atua na entrada *RESET* do "*FLIP-FLOP*" da seguinte maneira :

- Caso o elemento esteja energizado, o operador associado (Oxxxx ou Rxxxx) será desativado.
- Se o elemento estiver desenergizado, o operador associado (Oxxxx ou Rxxxx) mantém o valor corrente inalterado.

Este elemento é utilizado em conjunto com o elemento Bobina *SET*, descrito anteriormente.

8.3.4 Bobina Pulso (*ONE SHOT*)

Objetivo - O objetivo deste elemento é ativar a bobina de um relé por um **único ciclo** de varredura do programa. Esta ativação se dá através de uma transição de subida do sinal aplicado à sua entrada.

Modo de Inserção dos Elementos no programa

Posicionando-se o cursor na oitava coluna da linha desejada e teclando-se a letra G. Neste instante aparecerá um menu com sete opções, entre elas, temos “Mono Estável”. O usuário deve

selecionar a esta opção e teclar “Enter”. Este elemento necessita de no mínimo uma linha livre para a sua inserção.

Símbolo do Elemento

Parâmetros

- **P1** - identifica o relé que terá a sua bobina energizada. Este relé pode ser em memória (operador R) ou uma saída digital (operador O).

Entradas

- **E1** - entrada de ativação do bloco mono estável. Sempre que houver uma transição de subida nesta entrada, de desenergizada para energizada, o relé associado permanecerá ativado por um **único ciclo** de varredura do programa.

Saídas

- não tem.

Descrição - Este elemento está associado ao controle de saídas digitais (operadores Oxxxx) e contatos auxiliares (operadores Rxxxx).

8.3.5 Bobina Oscilador

Objetivo - O objetivo deste elemento é inverter o estado do contato a ele associado (P1) sempre que houver uma transição em sua entrada (E1), de desenergizado para energizado.

Modo de Inserção dos Elementos no programa

Posicionando-se o cursor na oitava coluna da linha desejada e teclando-se a letra G. Neste instante aparecerá um menu com sete opções, entre elas, temos “Oscilador”. O usuário deve selecionar a esta opção e teclar “Enter”. Este elemento necessita de no mínimo uma linha livre para a sua inserção.

Símbolo do Elemento

Parâmetros

- **P1** - identifica o contato que terá a sua bobina alterada. Este relé pode ser em memória (operador R) ou uma saída digital (operador O).

Entradas

- **E1** - entrada de controle do bloco Oscilador. Sempre que houver uma transição na entrada, de desenergizado para energizado, o contato associado terá seu valor corrente invertido.

Saídas

- não tem.

Descrição - Este elemento está associado ao controle de saídas digitais (operadores Oxxxx) e contatos auxiliares (operadores Rxxxx).

8.4 Movimentação de Dados

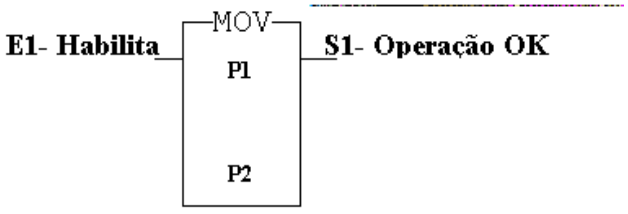
8.4.1 Movimentação Simples (MOV)

Objetivo - O objetivo deste elemento é transferir dados entre operadores, podendo estes operadores serem memórias, constantes, entradas e saídas analógicas.

Modo de Inserção dos Elementos no programa

Posicionando-se o cursor na posição desejada e teclando-se a letra M. Este elemento necessita de no mínimo duas linhas livres para a sua inserção.

Símbolo do Elemento



Parâmetros

- **P1** - representa um dos operadores da operação de movimentação de dados (MOV) e é a origem da movimentação. Pode ser uma memória inteira (operador M), uma constante inteira (operador K), uma memória real (operador D), uma constante real (operador Q), uma constante hexadecimal (operador H), uma entrada analógica (operador E) ou uma entrada digital (operador I).

Observação: se este parâmetro for especificado como uma entrada digital (operador I), ele deve corresponder ao primeiro canal do módulo de entrada digital (DIM), por exemplo, I0, I16, I32, etc.

- **P2** - representa um dos operadores da operação de movimentação de dados (MOV) e é o destino da movimentação. Pode ser uma memória inteira (operador M), uma memória real (operador D), uma saída analógica (operador S) ou uma saída digital (operador O).

Entradas

- **E1** - entrada de habilitação (Habilita) da operação de movimentação de dados (MOV). Sempre que esta entrada estiver energizada a operação se realizará, ou não, em caso contrário.

Saídas

- **S1** - esta saída segue o valor da entrada E1, ou seja, se E1 estiver energizada, S1 também estará, e vice e versa.

Descrição - Este elemento é representado por um bloco funcional com uma entrada, uma saída e tem dois operadores. O operador P1 pode ser uma constante inteira (operador K), real (operador Q) ou hexadecimal (H), uma memória inteira de 16 bits (operador M) ou real (operador D), uma entrada analógica (E) ou digital (I).

O operador P2 poder ser uma memória inteira (operador M) ou real (operador D) ou ainda uma saída analógica((S) ou digital (O). O conteúdo da memória/constante representada pelo operador superior (P1) é movido para o conteúdo da memória inteira/real ou saída digital/analógica representada pelo operador inferior (P2).

No caso dos operadores de entrada/saída analógica os valores movidos são sempre inteiros na faixa de 0 a 4095, refletindo o conteúdo de leitura/escrita das placas de entrada/saída analógica do MCI.

No caso dos operadores de entrada/saída digital os valores movidos são sempre inteiros, refletindo o conteúdo de leitura/escrita das placas de entrada/saída digital do MCI . Devem ser observadas as seguinte restrições em sua utilização, como mostrado na tabela a seguir:

Operador	Índice especificado	Quantidade de Valores Transferidos
I	0, 16, 32, ...	16 bits = inteiro
O	0, 16, 32, ...	16 bits = inteiro

Para o bloco de movimentação vale a seguinte regra de conversão automática de valores :

- Se o valor (P1) for movimentado para uma memória real, o mesmo será convertido para real, independentemente do valor de origem.

- Se o valor (P1) for movimentado para uma memória inteira, o mesmo será convertido para inteiro, independentemente do valor de origem.

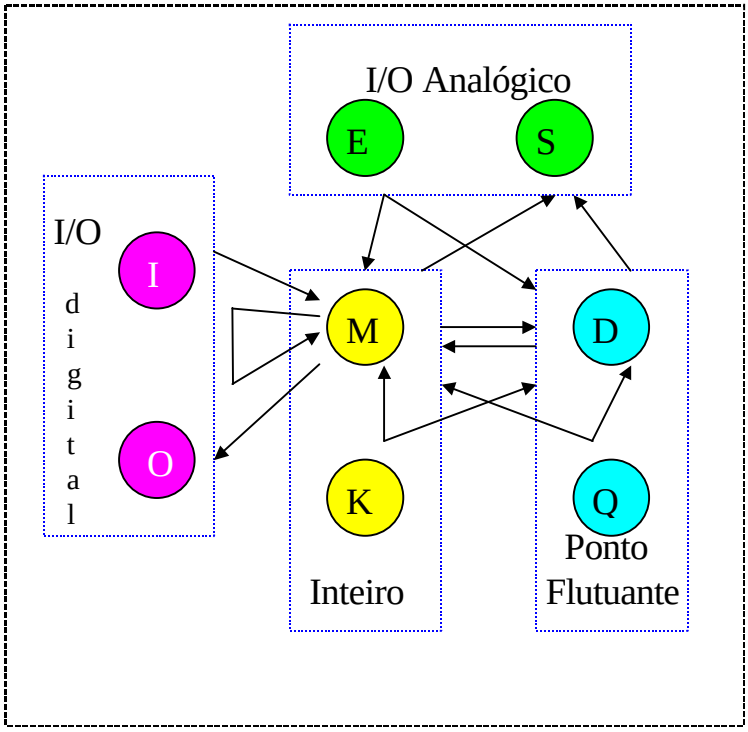


Figura - Possibilidades de Movimentação Simples

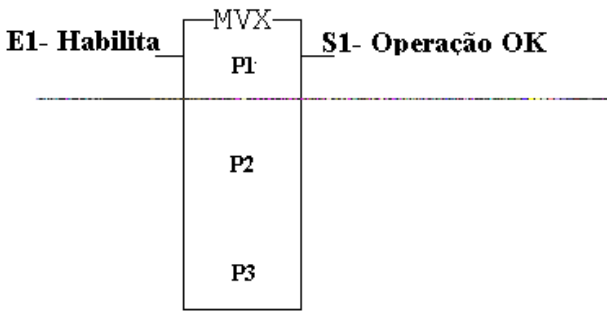
8.4.2 Movimentação Indexada (MVX)

Objetivo - Fazer a transferência de dados entre áreas de memória inteiras (operador M).

Modo de Inserção do Elemento no programa

Posicionando-se o cursor na posição desejada e teclando-se a letra E. Neste instante aparecerá um menu com quinze opções, entre elas, temos : “Movimentação Indexada”. O usuário deve selecionar a opção desejada e teclar “Enter”. Este elemento necessita de no mínimo três linhas livres para a sua inserção.

Símbolo do Bloco



Parâmetros

- **P1** - representa um dos operadores da operação de movimentação indexada de dados (MVX) e é a origem da movimentação. Pode ser uma memória inteira (operador M), uma constante inteira (operador K).

- **P2** - representa um dos operadores da operação de movimentação indexada de dados (MVX) e é o destino da movimentação. Pode ser uma memória inteira (operador M) ou uma constante inteira (operador K).

- **P3** - representa a quantidade de memórias inteiras que serão transferidas de uma só vez. Pode ser uma memória inteira (operador M) ou uma constante inteira (operador K).

Entradas

- **E1** - entrada de habilitação da operação de movimentação indexada de dados (MVX). Sempre que esta entrada estiver energizada a operação se realizará, ou não, em caso contrário.

Saídas

- **S1** - esta saída indica o sucesso ou não da operação de transferência de dados. Se estiver energizada, significa que a transferência de dados transcorreu com sucesso e que houve algum erro, caso esteja desenergizada.

Descrição - Este elemento é representado por um bloco funcional com uma entrada, uma saída e tem três operadores.

O operador P1 especifica o índice da memória ou a memória inicial que será transferida. O operador P2 especifica o índice da memória ou a memória inicial para onde serão transferidos os valores. O operador P3 especifica a quantidade de memórias que serão transferidas.

A saída S1 ("Erro") indica se a transferência pôde ser efetivada. Se estiver energizada é porque a transferência transcorreu corretamente e desenergizada caso contrário.

Exemplo de Utilização

Para transferir as memórias M10 a M15 para as memórias M20 a M25, os operadores devem ser especificados como mostrado a seguir.

Operador P1	Operador P2	Operador P3
M10 ou 10	M20 ou 20	6 ou M0

Neste exemplo, as seis memórias a partir de M10 até M15 serão transferidas para as memórias a partir de M20 até M25, seguindo a ordem crescente na transferência, ou seja, M10 será transferida para M20, M11 para M21, M12 para M22, M13 para M23, M14 para M24 e M15 para M25.

Na especificação do operador P3 foi colocado a memória M0 considerando que o conteúdo da mesma seja igual a 6, coincidindo com a constante especificada.

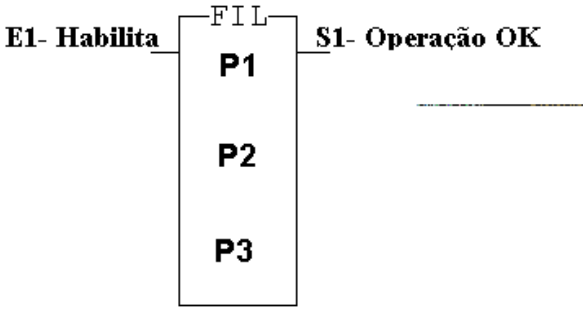
8.4.3 Inicialização de Dados (FIL)

Objetivo - Fazer a Inicialização de dados em área de memória inteira (operador M), memória real (operador D) ou contatos auxiliares (operador R).

Modo de Inserção do Elemento no programa

Posicionando-se o cursor na posição desejada e teclando-se a letra E. Neste instante aparecerá um menu com dezesseis opções, entre elas, temos : “Inicialização de dados”. O usuário deve selecionar esta opção e teclar “Enter”. Este elemento necessita de no mínimo três linhas livres para a sua inserção.

Símbolo do Bloco



Parâmetros

- **P1** - representa a variável inicial a ser inicializada. Pode ser uma memória inteira (operador M), ou uma constante inteira (operador K). Se por exemplo **P1** for especificado como M10 e M10 tiver o valor 5, então a variável a ser inicializada começará em M5 ou D5 ou R5. Se por exemplo **P1** for especificado como K20, então a variável a ser inicializada começará em M20 ou D20 ou R20.

- **P2** - representa o valor da Inicialização, podendo ser :

  - Se **P2** = R → T ou R
    Se o valor a ser usado para a Inicialização for um contato auxiliar, então este parâmetro pode ser o valor de um contato auxiliar (operador R) ou uma constante T0 ou T1. T0 representa um contato aberto e T1 um contato fechado. Se por exemplo **P2** for especificado como R10 e R10 estiver desligado, então os contatos das variáveis especificadas serão desligadas. Se por exemplo **P2** for especificado como T1, então os contatos das variáveis serão ligados e desligados, caso **P2** seja especificado como T0.

  - Se **P2** = M → K ou M
    Se o valor a ser usado para a Inicialização for uma memória inteira, então este parâmetro pode ser o valor de uma memória inteira (operador M) ou uma constante K. Se por exemplo **P2** for especificado como M10 e M10 tiver o valor 5, então as variáveis serão inicializadas com o valor 5. Se por exemplo **P2** for especificado como K20, então as variáveis serão inicializadas com o valor 20.

  - Se **P2** = D → Q ou D
    Se o valor a ser usado para a Inicialização for uma memória real, então este parâmetro pode ser o valor de uma memória real (operador D) ou uma constante Q. Se por exemplo **P2** for especificado como D10 e D10 tiver o valor 8.4, então as variáveis serão inicializadas com o valor 8.4. Se por exemplo **P2**

for especificado como Q7 e Q7 for igual a 3.5, então as variáveis serão inicializadas com o valor 3.5.

- **P3** - representa o número de variáveis a serem inicializadas a partir do parâmetro **P1**. Pode ser uma memória inteira (operador M) ou uma constante inteira (operador K). Se por exemplo **P3** for especificado como M10 e M10 tiver o valor 5, então as cinco variáveis a partir de **P1** serão inicializadas com o conteúdo especificado por **P2**. Se por exemplo **P3** for especificado como K20, então as vinte variáveis a partir de **P1** serão inicializadas com o conteúdo especificado por **P2**.

Entradas

- **E1** - entrada de habilitação (Habilita) da operação de Inicialização de dados . Sempre que esta entrada estiver energizada a operação se realizará, ou não, em caso contrário.

Saídas

- **S1** - esta saída indica o sucesso ou não da operação de Inicialização de dados. Se estiver energizada, significa que a Inicialização de dados transcorreu com sucesso e que houve algum erro, caso esteja desenergizada.

Descrição - Este elemento é representado por um bloco funcional com uma entrada, uma saída e tem três operadores. O operador **P1** pode ser uma memória inteira (operador M) ou uma constante inteira (operador K).

O operador **P2** poder ser uma constante inteira (operador K), uma memória inteira (operador M), uma memória real (operador D), uma constante real (operador Q), um contato auxiliar (operador R), uma constante T0 ou uma constante T1.

O parâmetro **P3** pode ser uma constante inteira (operador K) ou uma memória inteira (operador M). O parâmetro **P1** especifica o índice inicial do elemento que será inicializado. O parâmetro **P2** especifica o valor e o tipo de dados ser inicializado nos elementos. O parâmetro **P3** especifica a quantidade de elementos que serão inicializados.

A saída S1 indica se a Inicialização pôde ser efetivada. Se estiver energizada é porque a Inicialização transcorreu corretamente e desenergizada caso contrário.

Exemplo de Utilização

Para inicializar as memórias M10 a M15 com o conteúdo da memória M20, os operadores devem ser especificados como mostrado a seguir.

Operador P1	Operador P2	Operador P3
K10	M20	K6 ou M0

Neste exemplo, as seis memórias a partir de M10 até M15 serão inicializadas com o conteúdo da memória M20. Supondo que a memória M20 contenha o valor 53, então as memórias M10, M11, M12, M13, M14 e M15 serão inicializadas com este valor.

Na especificação do operador P3 foi colocado a memória M0 considerando que o conteúdo da mesma seja igual a 6, coincidindo com a constante especificada.

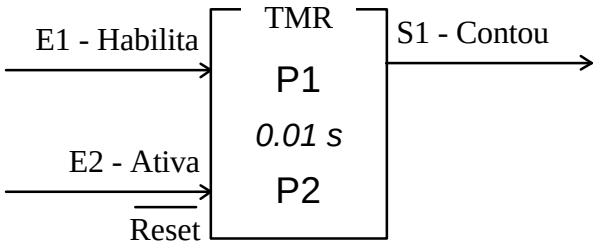
8.5 Temporizador (TMR)

Objetivo - Este elemento funciona como um temporizador. Uma vez carregado um determinado período de tempo como parâmetro e tendo a contagem sido habilitada, este valor é decrementado até que chegue a zero, quando a saída do bloco é ativada, indicando a situação de fim da contagem. O valor do temporizador é múltiplo de 0,01 segundos.

Modo de Inserção do Elemento no programa

Posicionando-se o cursor na posição desejada e teclando-se a letra T. Este elemento necessita de no mínimo duas linhas livres para a sua inserção.

Símbolo do Elemento



Parâmetros

- **P1** - representa o valor corrente da contagem do temporizador e deve ser obrigatoriamente uma memória inteira (operador M).
- **P2** - representa o valor inicial da contagem do elemento temporizador e dever ser obrigatoriamente uma memória inteira (operador M) ou constante inteira (operador K).

Entradas

- **E1** - entrada de habilitação (“Habilita”) do elemento temporizador. Quando estiver desenergizada, o temporizador está desabilitado (paralisado). Quando estiver energizada o temporizador está contando até que expire o tempo programado.
- **E2** - entrada de ativação (“Ativa/*Reset*”) do elemento temporizador. Quando estiver desenergizada, o temporizador está em seu estado de reset de contagem, não temporizando. Quando esta entrada estiver energizada, o temporizador está ativado, ou seja, apto a temporizar, dependendo apenas da entrada E1.

Saídas

- **S1** - Quando ativa indica fim da temporização (Expirou). Quando inativa indica que o temporizador está desabilitado ou desativado ou que ainda não expirou a temporização.

Descrição - Este elemento é representado por um bloco funcional com duas entradas (E1 e E2) e uma saída (S1) e utiliza dois operadores (P1 e P2). O operador P1, que aparece na parte superior do bloco, guarda o valor corrente da contagem do temporizador e deve ser obrigatoriamente um operador do tipo M. O operador P2 define o valor da temporização e pode ser um operador do tipo M ou uma constante K.

Quando a entrada E2 está desenergizada, fica caracterizado o estado de *Reset* do temporizador, independente do estado da entrada E1. Neste estado de *Reset*, o valor do parâmetro P2 é carregado no parâmetro P1.

Quando a entrada E2 estiver energizada, indica que o bloco do temporizador está ativo, e a entrada E1 indica a Habilitação do temporizador para a contagem. Neste caso, se E1 está ativa, o temporizador está apto a temporizar (temporizador contando), caso contrário, está paralisado a contagem do temporizador (temporizador paralisado). No estado de temporizador contando, o parâmetro P1 é decrementado até chegar a zero (temporizador expirado).

A saída S1 energizada indica que a temporização chegou ao final (temporizador expirado, indicando valor corrente da contagem de P1 igual a 0). Se o temporizador está desativado (E2 inativa) ou paralisado (E2 ativa, E1 ativa) ou contando (E2 ativa, E1 ativa e valor corrente de P1 diferente de 0) esta saída permanece desenergizada.

Se o temporizador estiver habilitado e contando, o valor corrente da contagem é decrementado de um a cada 0,01 segundos, até chegar a zero, quando então indica temporizador expirado. A tabela a seguir representa a operação do bloco temporizador.

E1 - Habilita	E2 - Ativa/Reset	Operação	Estado	S1 - Expirou
0	0	P1 ← P2	Reset	0
1	0	P1 ← P2	Reset	0
1	1	se P1 ≠ 0 decrementa P1	contando	0
1	1	se P1 = 0	expirado	1
0	1	mantém P1	paralisado	0

Na tabela de estado do temporizador, 0 (zero) representa um estado desenergizado, 1 (um) representa um estado energizado.

8.6 Contadores

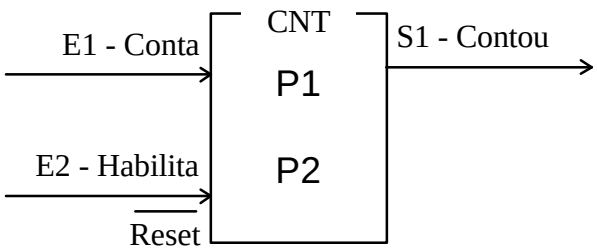
8.6.1 Contador Unidirecional (CNT)

Objetivo - O objetivo deste elemento é contar um determinado número de transições ocorridas na entrada "Conta". Ele conta o número de transições da entrada "Conta" até um certo limite de contagem fornecido pelo usuário como parâmetro. A sua saída indica o fim da contagem.

Modo de Inserção do Elemento no programa

Posicionando-se o cursor na posição desejada e teclando-se a letra C. Neste instante aparecerá um menu com duas opções : “Contador *Up*” e “Contador *Up/Down*”. O usuário deve selecionar a opção “Contador *Up*” e teclar “Enter”. Este elemento necessita de no mínimo duas linhas livres para a sua inserção.

Símbolo do Elemento



Parâmetros

- **P1** - representa o valor corrente da contagem do elemento e deve ser obrigatoriamente uma memória inteira (operador M).
- **P2** - representa o valor final (limite) da contagem do elemento e dever ser obrigatoriamente uma memória inteira (operador M) ou constante inteira (operador K).

Entradas

- **E1** - entrada de contagem (Conta) do elemento contador. Sempre que houver uma transição desta entrada de desenergizada para energizada, o contador terá o seu valor de contagem incrementado de um.
- **E2** - entrada de habilitação (Habilita) do elemento contador e enquanto estiver desenergizada, o contador estará desabilitado (sem contar).

Saídas

- **S1** - saída de indicação de fim da contagem (Contou) do elemento contador, ou que está desabilitado ou desativado.

Descrição - Este elemento é representado por um bloco funcional com duas entradas (E1 e E2) e uma saída (S1) e utiliza dois operadores (P1 e P2). O operador P1 que aparece na parte superior do bloco guarda o valor corrente da contagem e deve ser obrigatoriamente uma memória (M). O parâmetro P2 contêm o valor final da contagem, e pode ser uma memória (M) ou uma constante (K).

A entrada inferior (E2) habilita a contagem ("Habilita") e uma transição do estado desenergizado para o estado energizado nesta entrada faz com que o valor corrente da contagem passe para zero (reset do contador). A entrada superior (E1) representa a entrada de contagem ("Conta"), assim, para contar é necessário que a entrada "Habilita" (E2) esteja energizada e toda vez que a entrada "Conta" (E1) transitar de desenergizada para energizada o valor corrente da contagem será incrementado de um.

A saída (S1) indica se a contagem chegou ao final (contagem = valor final da contagem) ou se o contador está paralisado ou desabilitado (contagem diferente do valor final da contagem). A tabela representada a seguir a operação do contador unidirecional.

E1-Habilita	E2-Conta	Operação	Estado	S1 - Contou
0	0	não faz nada	parado	0
0→1	0	P1 ← 0	reset	0
1	0→1	P1 ← P1 + 1	contando	0
1	1	P1 = P2	contou	1

Na tabela de estado do contador unidirecional, 0 (zero) representa um estado desenergizado, 1 (um) representa um estado energizado, 0→1 representa uma transição do estado 0 para o estado 1.

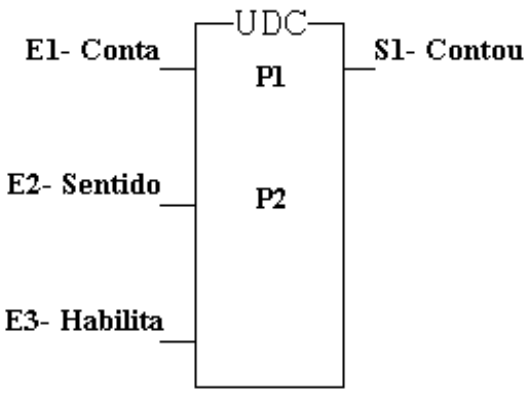
8.6.2 Contador Bidirecional (UDC)

Objetivo - O objetivo deste elemento é contar um determinado número de transições ocorridas na entrada "Conta". Ele conta o número de transições da entrada "Conta" até um certo limite de contagem fornecido pelo usuário como parâmetro. A contagem pode ser crescente (*Up*) ou decrescente (*Down*), dependendo do sentido fornecido pelo usuário como entrada para o bloco. A sua saída indica o fim da contagem.

Modo de Inserção do Elemento no programa

Posicionando-se o cursor na posição desejada e teclando-se a letra C. Neste instante aparecerá um menu com duas opções : "Contador *Up*" e "Contador *Up/Down*". O usuário deve selecionar a opção "Contador *Up*/*Down*" e teclar "Enter". Este elemento necessita de no mínimo três linhas livres para a sua inserção.

Símbolo do Elemento



Parâmetros

- **P1** - representa o valor corrente da contagem do elemento e deve ser obrigatoriamente uma memória inteira (operador M).
- **P2** - representa o valor final (limite) da contagem do elemento e dever ser obrigatoriamente uma memória inteira (operador M) ou constante inteira (operador K).

Entradas

- **E1** - entrada de contagem do elemento contador. Sempre que houver uma transição desta entrada de desenergizada para energizada, o contador terá o seu valor de contagem incrementado ou decrementado de um, dependendo da entrada E2 que indica o sentido da contagem.
- **E2** - entrada que define o sentido (Sentido) da contagem do elemento contador. Se estiver energizada o contador será sempre incrementado, e decrementado caso contrário.

- **E3** - entrada de habilitação (Habilita) do elemento contador e enquanto estiver desenergizada, o contador estará desabilitado (sem contar).

Saídas

- **S1** - saída de indicação de fim da contagem (Contou) do elemento contador, ou que está desabilitado ou desativado.

Descrição - Este elemento é representado por um bloco funcional com três entradas e uma saída e precisa de dois operadores. O operador que aparece na parte superior do bloco (P1) guarda o valor corrente do contador e deve ser obrigatoriamente uma memória (operador M) e o inferior (P2) que representa o valor final da contagem, pode ser uma memória ou uma constante.

A entrada inferior (E3) habilita a contagem ("Habilita") e quando esta transitar do estado desenergizado para o energizado, o valor corrente passará a ser zero se o sentido for 1 (energizado) ou passará a ser igual ao valor inicial da contagem caso contrário. A entrada do meio do bloco (E2) indica o sentido da contagem, ou seja, se esta entrada for igual a 1 o valor corrente da contagem é incrementado a cada transição da entrada "Conta" e será decrementado caso seja igual a 0. A entrada superior (E1) é a de contagem ("Conta").

Para contar é necessário que a entrada "Habilita" esteja energizada. Cada vez que a entrada "Conta" transitar de desenergizada para energizada o valor de contagem será decrementado ou incrementado de um, dependendo da entrada de sentido. A saída (S1) indica se a contagem chegou ao final (contagem = 0, se sentido = 0 e contagem = valor final, se sentido = 1) ou se o contador está parado ou desabilitado. A tabela a seguir representa a operação do contador unidirecional.

E3-Habilita	E1-Conta	Operação	Estado	S1-Contou
0	0	não incrementa	parado	0
0→1	0	valor corrente = 0	parado	0
1	0→1	increment. valor corrente	contando	0
1	1	valor corrente = valor final	parado	1

Tabela de estado do Contador Bidirecional com Sentido = 1

E3-Habilita	E1-Conta	Operação	Estado	S1-Contou
0	0	não decrementa	parado	0
0→1	0	valor corrente = valor inicial	parado	0
1	0→1	decrement. Valor corrente	contando	0
1	1	valor corrente = 0	parado	1

Tabela de estado do Contador Bidirecional com Sentido = 0

Nas tabelas de estado dos contadores, 0 (zero) representa um estado desenergizado, 1 (um) representa um estado energizado, 0→1 representa uma transição do estado 0 para o estado 1.

8.7 Funções Matemáticas

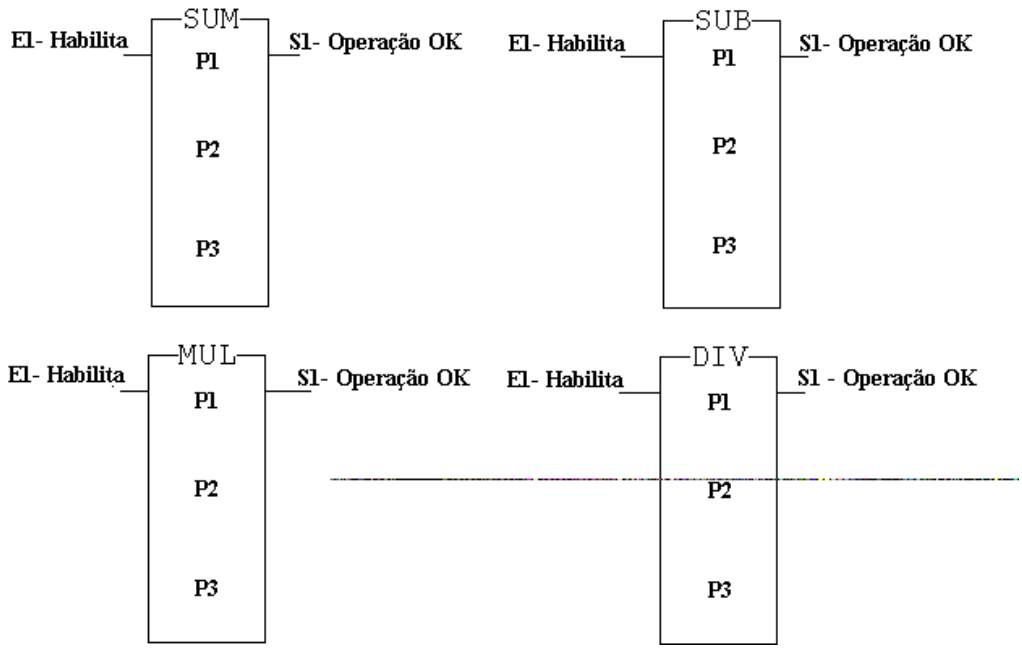
8.7.1 Operações Matemáticas (SUM, SUB, MUL, DIV)

Objetivo - O objetivo destes blocos é realizar as quatro operações aritméticas principais, soma, subtração, multiplicação e divisão.

Modo de Inserção dos Elementos no programa

Posicionando-se o cursor na posição desejada e teclando-se a letra O. Neste instante aparecerá um menu com dez opções, entre elas, temos : “Somador”, “Subtrator”, “Multiplicador” e “Divisor”. O usuário deve selecionar a opção desejada e teclar “Enter”. Estes elementos necessitam de no mínimo três linhas livres para a sua inserção.

Símbolos dos Elementos



Parâmetros

- **P1** - representa um dos operandos das operações matemáticas (Soma, Subtração, Multiplicação e Divisão). Deve ser obrigatoriamente uma memória inteira (operador M), uma constante inteira (operador K), uma memória real (operador D) ou uma constante real (operador Q).

- **P2** - representa um dos operandos das operações matemáticas (Soma, Subtração, Multiplicação e Divisão). Deve ser obrigatoriamente uma memória inteira (operador M), uma constante inteira (operador K), uma memória real (operador D) ou uma constante real (operador Q).

- **P3** - representa o resultado das operações matemáticas (Soma, Subtração, Multiplicação e Divisão). Deve ser obrigatoriamente uma memória inteira (operador M) ou uma memória real (operador D).

Entradas

- **E1** - entrada de habilitação (Habilita) das operações matemáticas (Soma, Subtração, Multiplicação e Divisão). Sempre que esta entrada estiver energizada as operações matemáticas se realizarão, ou não, em caso contrário.

Saídas

- **S1** - saída de indicação sucesso na operação matemática.

Descrição - Existem quatro blocos funcionais que implementam as quatro operações aritméticas. Estes elementos são representados por um bloco funcional com uma entrada e uma saída e precisa de três operadores. Os dois operadores que aparecem na parte superior do bloco (P1 e P2) guardam os valores que servirão como operandos nas operações e devem ser obrigatoriamente uma memória (M ou D) ou uma constante (K ou Q) e o inferior (P3) deve ser obrigatoriamente uma memória (M ou D) que conterá o valor resultante da operação. A entrada do bloco (E1) habilita a operação ("Habilita"). Para que a operação se realize é necessário que a entrada "Habilita" esteja energizada. A saída S1 indica se a operação aritmética foi executada com sucesso (saída TRUE) ou se houve erro na operação, como OVERFLOW (resultado superou capacidade de representação numérica da memória), divisão por zero, etc. (saída PALSE).

Para as quatro operações matemáticas vale a seguinte regra para a sua execução: O tipo de operação realizada dependerá sempre do primeiro operando (P1). Se ele for inteiro, a operação realizada será inteira. Se for real, a operação executada será real.

A atribuição do resultado das operações dependerá do tipo do terceiro operando (P3), como detalhado a seguir :

- Se o resultado for atribuído a uma memória real, o mesmo será convertido para real, independentemente do tipo de operação realizada.

- Se o resultado for atribuído a uma memória inteira, o mesmo será convertido para inteiro, independentemente do tipo de operação realizada.

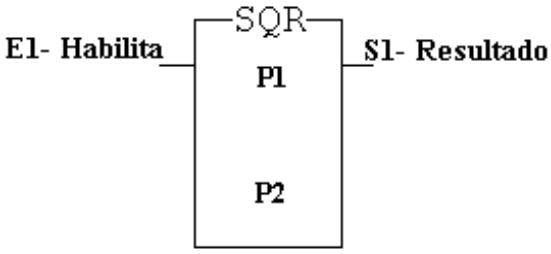
8.7.2 Raiz Quadrada (SQR)

Objetivo - O objetivo deste elemento é extrair a raiz quadrada de um determinado número fornecido como parâmetro.

Modo de Inserção do Elemento no programa

Posicionando-se o cursor na posição desejada e teclando-se a letra O. Neste instante aparecerá um menu com dez opções, entre elas, temos : “Raiz Quadrada”. O usuário deve selecionar a opção desejada e teclar “Enter”. Este elemento necessita de no mínimo duas linhas livres para a sua inserção.

Símbolo do Elemento



Parâmetros

- **P1** - representa o número real para o qual será extraída a sua raiz quadrada. Deve ser obrigatoriamente uma memória real (operador D) ou uma constante real (operador Q).
- **P2** - representa o resultado da operação de extração da raiz quadrada. Deve ser obrigatoriamente uma memória real (operador D).

Entradas

- **E1** - entrada de habilitação da operação de extração da raiz quadrada. Sempre que esta entrada estiver energizada a operação se realizará, ou não, em caso contrário.

Saídas

- **S1** - saída de indicação Resultado na operação matemática. Se operação OK retorna TRUE (energizado), caso contrário retorna FALSE (desenergizado).

Descrição - Este elemento é representado por um bloco funcional com uma entrada, uma saída e tem dois operadores. O operador P1 pode ser uma constante real (operador Q) ou uma memória real (operador D). O segundo parâmetro deve ser obrigatoriamente uma memória real (operador D). A entrada do bloco (E1) habilita a operação. Para que a operação se realize é necessário que a entrada "Habilita" esteja energizada. A saída (S1) indica se a operação foi executada corretamente ou não. Se ela não estiver energizada significa que a operação não se realizou corretamente e vice e versa. É importante salientar que a extração de raiz quadrada somente é realizada para valores reais.

8.8 Deslocamento à Direita (SHR)

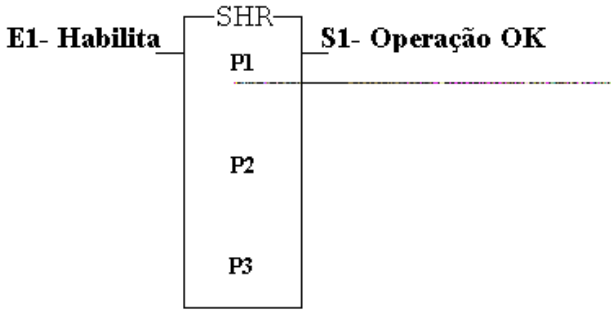
Objetivo - O objetivo deste elemento é realizar o deslocamento dos*bits* de um operador para a direita um número determinado de vezes.

Modo de Inserção dos Elementos no programa

Posicionando-se o cursor na posição desejada e teclando-se a letra O. Neste instante aparecerá um menu com dez opções, entre elas, temos : “Desl. Direita”. O usuário deve selecionar a

opção desejada e teclar “Enter”. Este elemento necessita de no mínimo três linhas livres para a sua inserção.

Símbolo do Elemento



Parâmetros

- **P1** - representa um dos operadores da operação de deslocamento à direita (SHR). Deve ser obrigatoriamente uma memória inteira (operador M).
- **P2** - representa um dos operadores da operação de deslocamento à direita (SHR). Deve ser obrigatoriamente uma memória inteira (operador M), uma constante inteira (operador K) ou uma constante hexadecimal (operador H).
- **P3** - representa o resultado da operação de deslocamento à direita (SHR). Deve ser obrigatoriamente uma memória inteira (operador M).

Entradas

- **E1** - entrada de habilitação da operação de deslocamento à direita (SHR). Sempre que esta entrada estiver energizada a operação se realizará, ou não, em caso contrário.

Saídas

- **S1** - esta saída segue o valor da entrada E1, ou seja, se E1 estiver energizada, S1 também estará, e vice e versa.

Descrição - Este elemento é representado por um bloco funcional com uma entrada, uma saída e tem três operadores. O operador P1 deve ser uma memória inteira (operador M). O segundo parâmetro pode ser uma memória inteira (operador M), uma constante inteira (operador K) ou uma constante hexadecimal (H). A entrada do bloco (E1) habilita a operação . Para que a operação se realize é necessário que a entrada (E1) esteja energizada.

O operador P1 é deslocado à direita o número de*bits* especificado pelo conteúdo do operador P2 e o resultado é colocado na memória representada pelo operador P3.

A saída (S1) é uma cópia da entrada, ou seja, se a entrada do bloco estiver energizada, a saída também estará. Vale a mesma regra para o caso de não estar energizada.

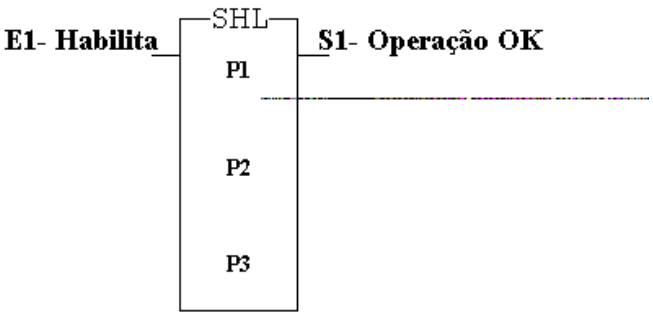
8.9 Deslocamento à Esquerda (SHL)

Objetivo - O objetivo deste elemento é deslocar os*bits* de um operador para a esquerda um determinado número de vezes.

Modo de Inserção dos Elementos no programa

Posicionando-se o cursor na posição desejada e teclando-se a letra O. Neste instante aparecerá um menu com dez opções, entre elas, temos : “Desl. Esquerda”. O usuário deve selecionar a opção desejada e teclar “Enter”. Este elemento necessita de no mínimo três linhas livres para a sua inserção.

Símbolo do Elemento



Parâmetros

- **P1** - representa um dos operadores da operação de deslocamento à esquerda (SHL). Deve ser obrigatoriamente uma memória inteira (operador M).
- **P2** - representa um dos operadores da operação de deslocamento à esquerda (SHL). Deve ser obrigatoriamente uma memória inteira (operador M), uma constante inteira (operador K) ou uma constante hexadecimal (operador H).
- **P3** - representa o resultado da operação de deslocamento à esquerda (SHL). Deve ser obrigatoriamente uma memória inteira (operador M).

Entradas

- **E1** - entrada de habilitação (Habilita) da operação de deslocamento à esquerda (SHL). Sempre que esta entrada estiver energizada a operação se realizará, ou não, em caso contrário.

Saídas

- **S1** - esta saída segue o valor da entrada E1, ou seja, se E1 estiver energizada, S1 também estará, e vice e versa.

Descrição - Este elemento é representado por um bloco funcional com uma entrada, uma saída e tem três operadores. O operador P1 deve ser uma memória inteira (operador M). O segundo parâmetro pode ser uma memória inteira (operador M), uma constante inteira (operador K) ou uma constante hexadecimal (H). A entrada do bloco (E1) habilita a operação ("Habilita"). Para que a operação se realize é necessário que a entrada "Habilita" esteja energizada.

O operador P1 é deslocado à esquerda o número de *bits* especificado pelo conteúdo do operador P2 e o resultado é colocado na memória representada pelo operador P3.

A saída (S1) é uma cópia da entrada, ou seja, se a entrada do bloco estiver energizada, a saída também estará. Vale a mesma regra para o caso de não estar energizada.

8.10 Comparações

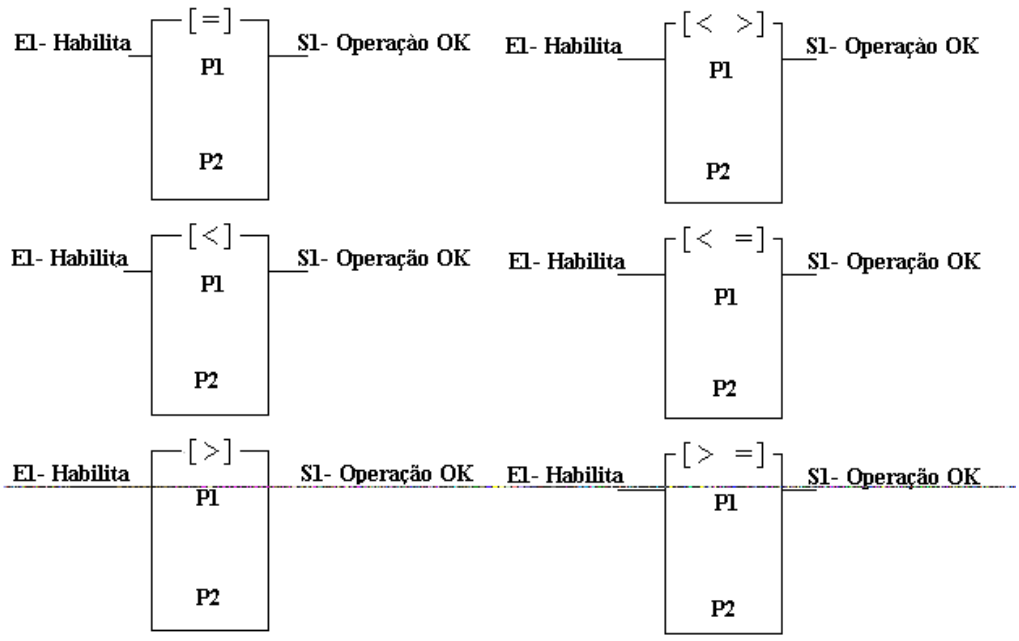
8.10.1 Comparações Matemáticas (=, <>, <, <=, > e >=)

Objetivo - O objetivo destes elementos é realizar a comparações entre operadores. Estas comparações podem ser : maior que, maior ou igual a, menor que, menor ou igual a e diferente.

Modo de Inserção dos Elementos no programa

Posicionando-se o cursor na posição desejada e teclando-se a letra P. Neste instante aparecerá um menu com sete opções, entre elas, temos : “[==]”, “[<>]”, “<]”, “[<=]”, “[>]”e “[>=]”. O usuário deve selecionar a opção desejada e teclar “Enter”. Estes elementos necessitam de no mínimo duas linhas livres para a sua inserção.

Símbolos dos Elementos



Parâmetros

- **P1** - representa um dos operadores das operações de comparação. Pode ser uma memória inteira (operador M) ou uma memória real (operador D).
- **P2** - representa um dos operadores das operações de comparação. Pode ser uma memória inteira (operador M), uma constante inteira (operador K), uma memória real (operador D), uma constante real (operador Q) ou uma constante hexadecimal (operador H).

Entradas

- **E1** - entrada de habilitação (Habilita) das operações de comparação. Sempre que esta entrada estiver energizada a operação se realizará, ou não, em caso contrário.

Saídas

- **S1** - esta saída indica o resultado das operações de comparação. Se estiver energizada, significa que a comparação é verdadeira e caso esteja desenergizada significa que a comparação é falsa.

Descrição - Através de três blocos de comparações disponíveis é possível obter-se as seguintes operações : "maior do que", "maior ou igual a", "menor do que", menor ou igual a" e "igual a". Os elementos de comparação são representados por um bloco funcional com uma entrada e uma saída e têm dois operadores. O primeiro operador (P1) tem que ser obrigatoriamente uma memória (M ou D) e o segundo (P2) pode ser uma memória (M ou D) ou uma constante (K, Q ou H). O valor da saída (S1) reflete o resultado da comparação, ou seja, energizada se a comparação for verdadeira e desenergizada caso contrário. Os blocos são apresentados a seguir.

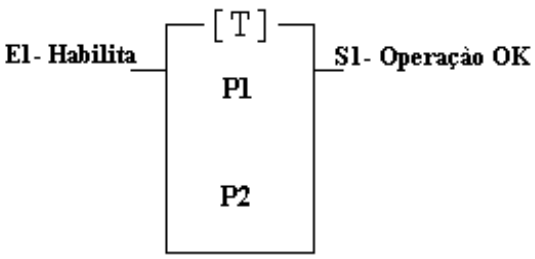
8.10.2 Teste Lógico *AND*

Objetivo - O objetivo deste elemento é realizar a operação lógica *AND* (E) *bit* a *bit* entre dois operadores.

Modo de Inserção do Elemento no programa

Posicionando-se o cursor na posição desejada e teclando-se a letra P. Neste instante aparecerá um menu com sete opções, entre elas, temos : “[&&]”. O usuário deve selecionar esta opção e teclar “Enter”. Este elemento necessita de no mínimo duas linhas livres para a sua inserção.

Símbolo do Elemento



Parâmetros

- **P1** - representa um dos operadores da função E (*AND*) *bit* a *bit*. Deve ser obrigatoriamente uma memória inteira (operador M).

- **P2** - representa um dos operadores da operação lógica E (*AND*) *bit* a *bit*. Deve ser obrigatoriamente uma memória inteira (operador M) ou uma constante inteira (operador K) ou uma constante hexadecimal (operador H).

Entradas

- **E1** - entrada de habilitação (Habilita) da operação lógica E (*AND*) *bit* a *bit*. Sempre que esta entrada estiver energizada a operação se realizará, ou não, em caso contrário.

Saídas

- **S1** - esta saída indica o resultado da operação. Se energizada, o resultado do teste lógico foi diferente de zero, desenergizada caso contrário.

Descrição - Este elemento é representado por um bloco funcional com uma entrada, uma saída e tem dois operadores. O operador P1 deve ser uma memória inteira (operador M). O segundo parâmetro pode ser uma memória inteira (operador M), uma constante inteira (operador K) ou uma constante hexadecimal (H). A entrada do bloco (E1) habilita a operação ("Habilita"). Para que a operação se realize é necessário que a entrada "Habilita" esteja energizada.

É realizada uma operação lógica *AND bit* a *bit* entre os operadores P1 e P2 e o resultado é refletido na saída S1.

A saída S1 indica o resultado da operação. Se energizada, o resultado do teste lógico foi diferente de zero, desenergizada caso contrário.

8.11 Operações Lógicas

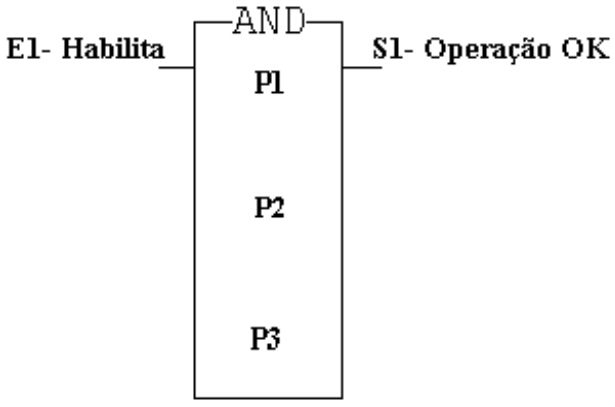
8.11.1 Operador *AND*

Objetivo - O objetivo deste elemento é realizar a operação lógica *AND* (E) entre dois operadores.

Modo de Inserção do Elemento no programa

Posicionando-se o cursor na posição desejada e teclando-se a letra O. Neste instante aparecerá um menu com dez opções, entre elas, temos : “Operador *AND*”. O usuário deve selecionar a opção desejada e teclar “Enter”. Este elemento necessita de no mínimo três linhas livres para a sua inserção.

.Símbolo do Elemento



Parâmetros

- **P1** - representa um dos operadores da operação lógica E (*AND*). Deve ser obrigatoriamente uma memória inteira (operador M).

- **P2** - representa um dos operadores da função E (*AND*). Deve ser obrigatoriamente uma memória inteira (operador M), uma constante inteira (operador K) ou uma constante hexadecimal (operador H).

- **P3** - representa o resultado da operação lógica E (*AND*). Deve ser obrigatoriamente uma memória inteira (operador M).

Entradas

- **E1** - entrada de habilitação da operação lógica E (*AND*). Sempre que esta entrada estiver energizada a operação se realizará, ou não, em caso contrário.

Saídas

- **S1** - esta saída segue o valor da entrada E1, ou seja, se E1 estiver energizada, S1 também estará, e vice e versa.

Descrição - Este elemento é representado por um bloco funcional com uma entrada, uma saída e tem três operadores. O operador P1 deve ser uma memória inteira (operador M).

O segundo parâmetro pode ser uma memória inteira (operador M), uma constante inteira (operador K) ou uma constante hexadecimal (H). A entrada do bloco (E1) habilita a operação. Para que a operação se realize é necessário que a entrada (E1) esteja energizada.

É realizada uma operação lógica *AND* entre os operadores P1 e P2 e o resultado é colocado na memória representada pelo operador P3.

A saída (S1) é uma cópia da entrada, ou seja, se a entrada do bloco estiver energizada, a saída também estará. Vale a mesma regra para o caso de não estar energizada.

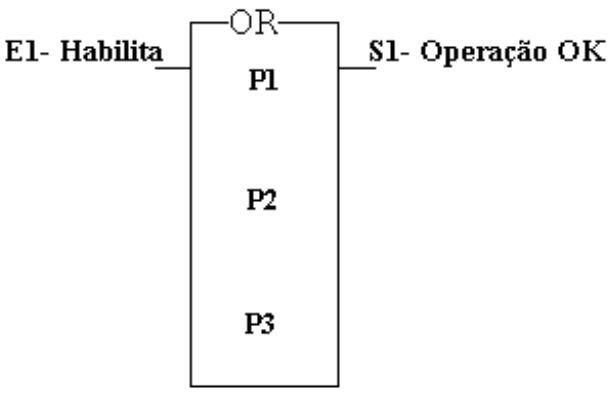
8.11.2 Operador *OR*

Objetivo - O objetivo deste elemento é realizar uma operação lógica *OR* (ou) entre dois operadores.

Modo de Inserção do Elemento no programa

Posicionando-se o cursor na posição desejada e teclando-se a letra O. Neste instante aparecerá um menu com dez opções, entre elas, temos : “Operador *OR*”. O usuário deve selecionar a opção desejada e teclar “Enter”. Este elemento necessita de no mínimo três linhas livres para a sua inserção.

Símbolo do Elemento



Parâmetros

- **P1** - representa um dos operadores da operação lógica OU (*OR*). Deve ser obrigatoriamente uma memória inteira (operador M)
- **P2** - representa um dos operadores da função OU (*OR*). Deve ser obrigatoriamente uma memória inteira (operador M), uma constante inteira (operador K) ou uma constante hexadecimal (operador H).
- **P3** - representa o resultado da operação lógica OU (*OR*). Deve ser obrigatoriamente uma memória inteira (operador M).

Entradas

- **E1** - entrada de habilitação da operação lógica OR (*OR*). Sempre que esta entrada estiver energizada a operação se realizará, ou não, em caso contrário.

Saídas

- **S1** - esta saída segue o valor da entrada E1, ou seja, se E1 estiver energizada, S1 também estará, e vice e versa.

Descrição - Este elemento é representado por um bloco funcional com uma entrada, uma saída e tem três operadores. O operador P1 deve ser uma memória inteira (operador M). O segundo parâmetro pode ser uma memória inteira (operador M), uma constante inteira (operador K) ou uma constante hexadecimal (H). A entrada do bloco (E1) habilita a operação . Para que a operação se realize é necessário que a entrada (E1) esteja energizada.

É realizada uma operação lógica *OR* entre os operadores P1 e P2 e o resultado é colocado na memória representada pelo operador P3.

A saída (S1) é uma cópia da entrada, ou seja, se a entrada do bloco estiver energizada, a saída também estará. Vale a mesma regra para o caso de não estar energizada.

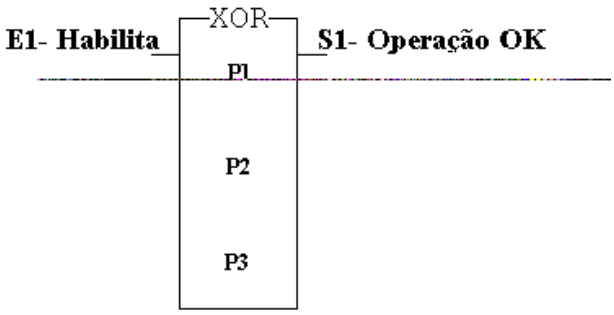
8.11.3 Operador *XOR*

Objetivo - O objetivo deste elemento é realizar uma operação lógica *XOR* (ou exclusivo) entre dois operadores.

Modo de Inserção do Elemento no programa

Posicionando-se o cursor na posição desejada e teclando-se a letra O. Neste instante aparecerá um menu com dez opções, entre elas, temos : "Operador *XOR*". O usuário deve selecionar a opção desejada e teclar "Enter". Este elemento necessita de no mínimo três linhas livres para a sua inserção.

Símbolo do Elemento



Parâmetros

- **P1** - representa um dos operadores da operação lógica OU EXCLUSIVO (*XOR*). Deve ser obrigatoriamente uma memória inteira (operador M).
- **P2** - representa um dos operadores da função OU EXCLUSIVO (*XOR*). Deve ser obrigatoriamente uma memória inteira (operador M), uma constante inteira (operador K) ou uma constante hexadecimal (operador H).
- **P3** - representa o resultado da operação lógica OU EXCLUSIVO (*XOR*). Deve ser obrigatoriamente uma memória inteira (operador M).

Entradas

- **E1** - entrada de habilitação da operação lógica OU EXCLUSIVO (*XOR*). Sempre que esta entrada estiver energizada a operação se realizará, ou não, em caso contrário.

Saídas

- **S1** - esta saída segue o valor da entrada E1, ou seja, se E1 estiver energizada, S1 também estará, e vice e versa.

Descrição - Este elemento é representado por um bloco funcional com uma entrada, uma saída e tem três operadores. O operador P1 deve ser uma memória inteira (operador M). O segundo parâmetro pode ser uma memória inteira (operador M), uma constante inteira (operador K) ou uma constante hexadecimal (H). A entrada do bloco (E1) habilita a operação . Para que a operação se realize é necessário que a entrada (E1) esteja energizada.

É realizada uma operação lógica *XOR* entre os operadores P1 e P2 e o resultado é colocado na memória representada pelo operador P3.

A saída (S1) é uma cópia da entrada, ou seja, se a entrada do bloco estiver energizada, a saída também estará. Vale a mesma regra para o caso de não estar energizada.

8.12 Blocos de Controle de Programa

8.12.1 Fim de Programa (ENDPG)

Objetivo - O objetivo deste elemento é delimitar o final da área do programa principal da aplicação ladder. Este bloco sempre deve estar presente em um programa, sinalizando o fim do mesmo.

Modo de Inserção dos Elementos no programa

Posicionando-se o cursor na oitava coluna da linha desejada e teclando-se a letra G. Neste instante aparecerá um menu com sete opções, entre elas, temos “Fim de Programa”. O usuário deve selecionar a esta opção e teclar “Enter”. Este elemento necessita de no mínimo uma linha livre para a sua inserção.

Símbolo do Elemento

Parâmetros

- não tem.

Entradas

- não tem.

Saídas

- não tem.

Descrição - Este elemento é representado por um bloco sem entrada. Este elemento funciona como um delimitador de área de execução do programa, ou seja, o ponto do programa em que se encontra delimita a área de execução do programa, não possuindo qualquer outra funcionalidade. Este elemento deve obrigatoriamente ocupar a oitava coluna da janela de edição, devendo ser a última linha de um programa.

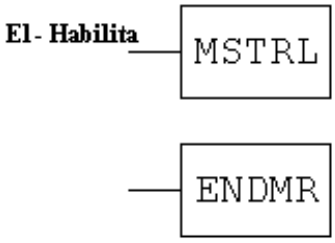
8.12.2 Inicio Relé Mestre / Fim de Relé Mestre

Objetivo - O objetivo destes elemento é delimitar uma área do programa que será executada sempre que o bloco **Relé Mestre** estiver ativado. Basicamente o bloco **Relé Mestre** tem dupla utilidade, pois marca o início da área a ser executada e ao mesmo tempo permite que esta mesma área do programa seja executada quando o bloco estiver energizado. Este bloco utiliza o bloco Fim de Relé Mestre como delimitador final da área de programa que deverá ser executada. Um programa pode ter quantos relés mestre se desejar.

Modo de Inserção dos Elementos no programa

Posicionando-se o cursor na oitava coluna da linha desejada e teclando-se a letra G. Neste instante aparecerá um menu com sete opções, entre elas, temos : “Relé Mestre” e “Fim de Relé Mestre”. O usuário deve selecionar a opção desejada e teclar “Enter”. Estes elementos necessitam de no mínimo uma linha livre para a sua inserção.

Símbolo do Elemento



Parâmetros

- **Relé Mestre** - não tem.
- **Fim de Relé Mestre** - não tem.

Entradas

- **Relé Mestre** - **E1** - entrada de habilitação (Habilita) do bloco de **Relé Mestre**. Sempre que esta entrada estiver energizada a área do programa, compreendida entre este bloco e o bloco de **Fim de Relé Mestre**, será executada.
- **Fim de Relé Mestre** - não tem.

Saídas

- não tem.

Descrição

- **Relé Mestre** - Este elemento é representado por um bloco de uma entrada. Este elemento funciona como uma bobina, ou seja, quando tiver energizada a sua entrada (**E1**) a área de programa delimitada por ele e o bloco **Fim de Relé Mestre** será executada, caso contrário esta área do programa não será executada. Este elemento deve obrigatoriamente ocupar a oitava coluna da janela de edição.

- **Fim de Relé Mestre** - Este elemento é representado por um bloco sem entrada. Este elemento funciona como um delimitador de área de programa, ou seja, o ponto do programa em que se encontra delimita a área de atuação do **Relé Mestre** imediatamente anterior a ele, não possuindo qualquer outra funcionalidade. Este elemento deve obrigatoriamente ocupar a oitava coluna da janela de edição.

Observação importante: Se na área delimitada pelo **Relé Mestre** e **Fim de Relé Mestre** existir alguma bobina e o **Relé Mestre** estiver desenergizado, então as bobinas estarão resetadas (desligadas), independentemente do estado das lógicas associadas a elas.

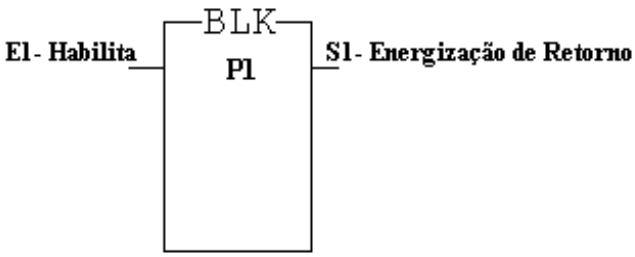
8.12.3 Bloco de Lógica (Subrotinas - BLK)

Objetivo - O objetivo deste elemento é executar o código de uma área do programa delimitada pelos blocos **Início de Bloco de Lógica** e **Fim de Bloco de Lógica**, sempre que este bloco estiver ativado. Um programa pode ter quantos Blocos de Lógica se desejar. Fazendo uma analogia com os termos de programação, este bloco seria uma chamada de uma rotina, ao passo que o bloco **Início de Lógica** corresponderia ao delimitador de início da rotina e o bloco de **Fim de Lógica** o seu delimitador de fim de rotina.

Modo de Inserção dos Elementos no programa

Posicionando-se o cursor na oitava coluna da linha desejada e teclando-se a letra G. Neste instante aparecerá um menu com sete opções, entre elas, temos : “Bloco de Lógica”. O usuário deve selecionar esta opção e teclar “Enter”. Este elemento necessita de no mínimo duas linhas livres para a sua inserção.

Símbolo do Elemento



Parâmetros

- **Txxxx** - identifica o **Bloco de Lógica** que será executado quando a entrada E1 estiver energizada. Este valor tem que ser um valor maior ou igual a zero. Quando este valor é zero, o **Bloco de Lógica** será executado automaticamente toda vez em que o controlador for ligado, servindo como uma rotina de Inicialização do programa. De outra forma, o **Bloco de Lógica** será executado apenas quando estiver a sua entrada energizada.

Entradas

- **E1** - entrada de habilitação do **Bloco de Lógica**. Sempre que esta entrada estiver energizada o **Bloco de Lógica** identificado pelo parâmetro Txxxx estará sendo executado, ou não, em caso contrário.

Saídas

- **S1** - esta saída segue o valor da energização da linha em que se encontra o bloco **Fim de Bloco de Lógica**. Se esta linha estiver energizada, S1 também estará, e vice e versa.

Descrição - Este elemento é representado por um bloco com uma entrada, uma saída e um parâmetro. Este elemento funciona como habilitação de execução e um **Bloco de Lógica** delimitado

pelos blocos **Início de Bloco de Lógica** e **Fim de Bloco de Lógica**. Quando energizado, o **Bloco de Lógica** identificado pelo parâmetro Txxxx será executado, ou não, caso contrário.

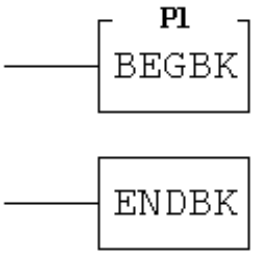
8.12.4 Início/Fim de Bloco de Lógica (Subrotinas - BEGBK e ENDBK)

Objetivo - O objetivo destes elementos é delimitar o início e o fim de uma área do programa que será executada sempre que o **Bloco de Lógica** estiver ativado. Estes blocos sempre devem ser utilizados em conjunto. O bloco **Início de Bloco de Lógica** delimita o início e o bloco **Fim de Bloco de Lógica** delimita o fim da área de programa que deverá ser executada quando o **Bloco de Lógica** correspondente estiver energizado. Um programa pode ter quantos Blocos de Lógica se desejar. Fazendo uma analogia com os termos de programação, o **Bloco de Lógica** seria uma chamada de uma rotina, ao passo que o bloco **Início de Lógica** corresponderia ao delimitador de início da rotina e o bloco de **Fim de Lógica** o seu delimitador de fim de rotina.

Modo de Inserção dos Elementos no programa

Posicionando-se o cursor na oitava coluna da linha desejada e teclando-se a letra G. Neste instante aparecerá um menu com sete opções, entre elas, temos : “Início” e “Fim de Bloco de Lógica” O usuário deve selecionar a opção desejada e teclar “Enter”. Estes elementos necessitam de no mínimo uma linha livre para a sua inserção.

Símbolo do Elemento



Parâmetros

- **Início de Bloco de Lógica**
  Txxxx - identifica o **Bloco de Lógica** que este bloco está delimitando o início. Este valor tem que ser um valor maior ou igual a zero. Quando este valor é zero, o Bloco de Lógica será executado automaticamente toda vez em que o controlador for ligado, servindo como uma rotina de Inicialização do programa. Quando este valor for diferente de zero, o bloco de Lógica será executado apenas quando o Bloco de Lógica correspondente estiver energizado.

- **Fim de Bloco de Lógica**
  não tem

Entradas

- **Início de Bloco de Lógica**
  não tem.

- **Fim de Bloco de Lógica**

não tem

Saídas

- **Início de Bloco de Lógica**
  não tem.

- **Fim de Bloco de Lógica**
  - não tem. Entretanto, este bloco retorna a energização da linha em que se encontra para o **Bloco de Lógica**. A saída do **Bloco de Lógica** corresponderá à energização da linha deste bloco. Se estiver energizada, a saída do **Bloco de Lógica** também estará energizada, ou não, caso contrário.

Descrição

- **Início de Bloco de Lógica**
  - Este elemento é representado por um bloco sem entrada. Este elemento funciona como um delimitador de área de programa, ou seja, o ponto do programa em que se encontra delimita o início da área de atuação do **Bloco de Lógica** que ele representa, não possuindo qualquer outra funcionalidade. Este elemento deve obrigatoriamente ocupar a oitava coluna da janela de edição e deve ser posicionado sempre após o delimitador de fim de programa (bloco de **Fim de Programa**).

- **Fim de Bloco de Lógica**
  - Este elemento é representado por um bloco sem entrada. Este elemento funciona como um delimitador de área de programa, ou seja, o ponto do programa em que se encontra delimita o fim da área de atuação do **Bloco de Lógica** que ele representa, não possuindo qualquer outra funcionalidade. Este elemento deve obrigatoriamente ocupar a oitava coluna da janela de edição e deve ser posicionado sempre após o delimitador de fim de programa (bloco de **Fim de Programa**) e também após o delimitador de **Início de Bloco de Lógica** correspondente.

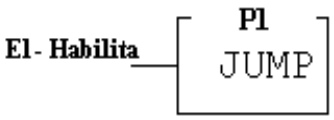
8.12.5 Bobina de Salto (JUMP)

Objetivo - O objetivo deste elemento é não executar um determinado número de lógicas caso tenha a sua entrada energizada. Ele não executa um determinado número de lógicas pulando-as. Este pulo é passado como parâmetro para o bloco.

Modo de Inserção dos Elementos no programa

Posicionando-se o cursor na oitava coluna da linha desejada e teclando-se a letra G. Neste instante aparecerá um menu com sete opções, entre elas, temos : “Bobina de Salto”. O usuário deve selecionar esta opção e teclar “Enter”. Este elemento necessita de no mínimo uma linha livre para a sua inserção.

Símbolo do Elemento



Parâmetros

- **Txxxx** - identifica a lógica para a qual o programa irá saltar, ou seja, passará a executar. Este valor tem que ser maior do que o número da lógica corrente e menor ou igual à última lógica.

Entradas

- **E1** - entrada de habilitação (Habilita) do bloco de salto. Sempre que esta entrada estiver energizada o programa passará a executar a partir da lógica indicada como parâmetro, saltando as lógicas que estiverem no intervalo entre a lógica corrente e a especificada.

Saídas

- não tem.

Descrição - Este elemento é representado por um bloco de uma entrada com um operador. O operador (Txxxx) especifica o número da lógica para onde deseja-se saltar. Este elemento funciona como uma bobina, ou seja, quando tiver energizada a sua entrada (E1) o programa saltará para a lógica especificada pelo operador, caso contrário o programa continuará a ser executado na lógica seguinte à bobina de salto. Não é permitido saltar para uma lógica anterior à bobina de salto. Este elemento deve obrigatoriamente ocupar a oitava coluna da janela de edição.

8.13 Interface Homem Máquina

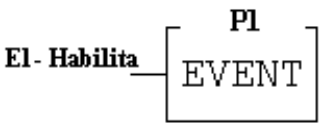
8.13.1 Bloco de Sinalização de Eventos (EVENT)

Objetivo - O objetivo deste elemento é sinalizar, através da interface homem/máquina do MCI, a ocorrência de algum evento.

Modo de Inserção dos Elementos no programa

Posicionando-se o cursor na oitava coluna da linha desejada e teclando-se a letra G. Neste instante aparecerá um menu com sete opções, entre elas, temos “Bloco de Sinalização de Eventos”. O usuário deve selecionar a esta opção e teclar “Enter”. Este elemento necessita de no mínimo uma linha livre para a sua inserção.

Símbolo do Elemento



Parâmetros

- **P1** - identifica a mensagem de sinalização de eventos que será mostrada no *display* do MCI enquanto este elemento estiver energizado.

Entradas

- **E1** - entrada de habilitação (Habilita) do bloco de sinalização de eventos (*EVENT*). Sempre que esta entrada estiver energizada a mensagem associada será exibida no *display* do MCI.

Saídas

- não tem.

Descrição - Este elemento é representado por um bloco de uma entrada e possui um operador. O operador (P1) especifica o número da mensagem de sinalização de eventos que aparecerá no *display* da interface homem/máquina do MCI. Se a sua entrada (E1) estiver energizada, a mensagem de sinalização de evento especificada pelo usuário será mostrada no *display*. Esta mensagem permanecerá no *display* enquanto a entrada do bloco estiver energizada. Uma outra mensagem pode ser colocada no *display* no lugar da mensagem corrente. Esta situação pode ocorrer se a mensagem de outro bloco for de maior prioridade do que a do bloco que estiver sendo mostrado. A especificação das mensagens é feita através da opção **ALT T** no editor de diagramas de contatos. A mensagem de maior prioridade é de número 0 (zero), e assim sucessivamente até o número de mensagens do sistema. Este bloco deve ser obrigatoriamente inserido na oitava coluna da janela de edição do editor de diagramas de contatos, na mesma posição reservada para as bobinas.

Observação: Este bloco somente pode ser utilizado se o MCI02 estiver configurado com o módulo KBD240, que é a interface com o usuário (conjunto teclado/*display*) ou com o módulo de interface remota ROP480.

Observação: Este bloco somente pode ser utilizado se o MCI03 estiver configurado com o módulo de interface remota ROP480, que é a interface com o usuário (conjunto teclado/*display*).

8.14 *Blocos Especiais*

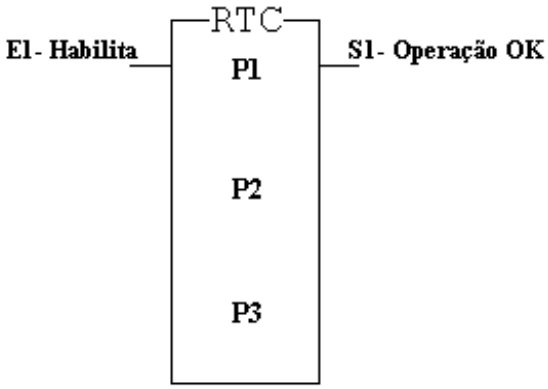
8.14.1 Relógio de Tempo Real (RTC)

Objetivo - O objetivo deste elemento é ter acesso ao relógio de tempo real do MCI. Este relógio inclui a data (dia, mês e ano) e hora (hora, minutos e segundos) correntes.

Modo de Inserção do Elemento no programa

Posicionando-se o cursor na linha desejada e teclando-se a letra E. Neste instante aparecerá um menu com dezesseis opções. O usuário deve selecionar a opção “Relógio de Tempo Real” e teclar “Enter”. Este elemento necessita de no mínimo três linhas livres para a sua inserção.

Símbolo do Elemento



Parâmetros

- **P1** - representa uma memória inteira (operador M) na qual serão atualizados os segundos e o dia da semana. Os segundos são codificados nos *bits* de 0 a 7 e o dia da semana é codificado nos *bits* de 8 a 15.
- **P2** - representa uma memória inteira (operador M) na qual serão atualizados os minutos e a hora correntes. Os minutos são codificados nos *bits* de 0 a 7 e a hora é codificada nos *bits* de 8 a 15.
- **P3** - representa uma memória inteira (operador M) na qual serão atualizados o dia, mês e ano do calendário. O dia é codificado nos *bits* de 0 a 5, o mês nos *bits* de 6 a 9 e o ano nos *bits* de 10 a 15. Note-se que o ano refere-se a um offset à partir do ano de 1990.

Entradas

- **E1** - Entrada de atualização do relógio/calendário. Os dados do relógio/calendário são atualizados enquanto esta entrada estiver energizada.

Saídas

- **S1** - Saída de indicação de relógio de tempo real. Esta saída, quando energizada, indica que o relógio de tempo real está presente no módulo de controle (CPU) do MCI, e ausente caso contrário.

Descrição - Este elemento se destina à contagem de tempo e funciona como um relógio calendário, podendo ser consultado o tempo corrente. O tempo inclui a data com o dia, mês e ano e o tempo com a hora, os minutos e os segundos. Para funcionar, este elemento depende de uma memória especial que tem embutida um relógio de tempo real.

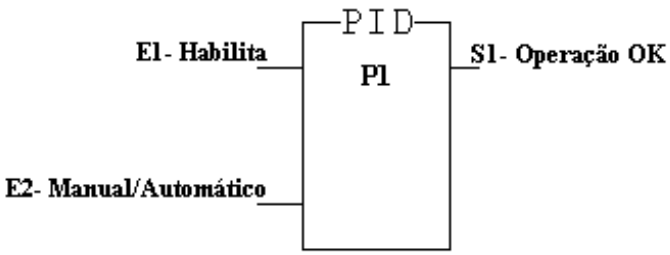
8.14.2 Controle PID

Objetivo - O objetivo deste bloco é executar um algoritmo de controle Proporcional, Integral e Derivativo (PID).

Modo de Inserção do Elemento no programa

Posicionando-se o cursor na posição desejada e teclando-se a letra E. Neste instante aparecerá um menu com quinze opções, entre elas, temos : “Controlador PID”. O usuário deve selecionar esta opção e teclar “Enter”. Estes elemento necessita de no mínimo duas linhas livres para a sua inserção.

Símbolo do Elemento



Parâmetros

- **P1** - identifica o bloco de controle. Pode estar na faixa entre 0 a 3, já que quatro blocos é o número máximo de blocos de controle PID permitidos em um programa.

Entradas

- **E1** - entrada de habilitação (Habilita) do bloco de controle PID. Sempre que esta entrada estiver energizada o controle estará sendo executado, ou não, em caso contrário.

- **E2** - entrada de modo de operação (Manual/Automático) do bloco de controle PID. Sempre que esta entrada estiver energizada o controle estará sendo executado em modo automático, e em modo manual caso contrário (veja no item "Descrição" as implicações do modo manual e automático).

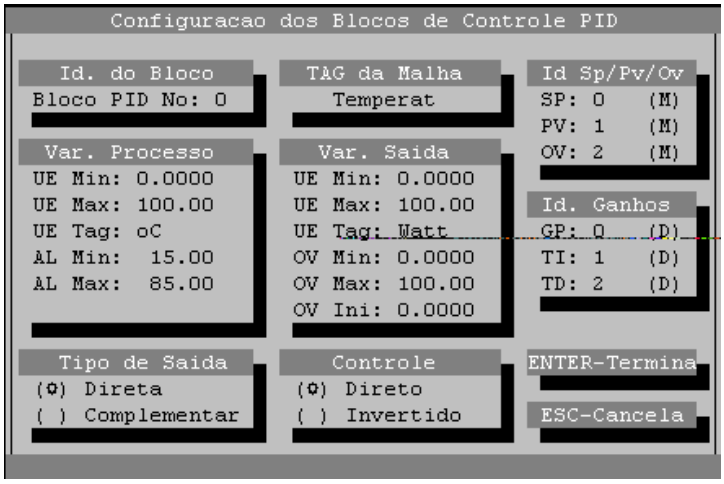
Saídas

- **S1** - esta saída segue o valor da entrada E1, ou seja, se E1 estiver energizada, S1 também estará, e vice e versa.

Descrição - Este elemento é representado por um bloco de duas entradas e uma saída, com um operador. O operador (P1) especifica o número do bloco de controle PID, de 0 a 3. Os parâmetros de controle são inseridos durante a fase de edição do programa através do comando "Blocos Contr." da janela de comandos.

O estado da entrada do bloco (E1) será refletido na saída (S1) sem qualquer alteração. Se a entrada deste bloco estiver energizada, a malha de controle estará operando em modo AUTOMÁTICO e em modo MANUAL caso contrário. No modo automático o bloco de controle estará atuando diretamente na variável de saída da malha PID, enquanto no modo manual o último valor da saída será mantido constante.

O bloco de controle PID para ser utilizado precisa de vários parâmetros que devem ser fornecidos pelo usuário. Para ter-se acesso aos parâmetros de controle do bloco PID é necessário posicionar o cursor do editor em cima do bloco PID e teclar-se**ALT** e **T**. Através deste comando, têm-se acesso à janela de "Configuração dos Blocos de Controle PID". A especificação dos parâmetros deve ser feita para todos os blocos inseridos no programa do usuário.



Na tela de configuração de blocos de controle PID existem algumas janelas em que podem ser especificados parâmetros, como descrito a seguir:

Id. do Bloco

Nesta janela é mostrado o número do bloco PID que está sendo configurado, servindo apenas como referência para o usuário saber para qual bloco de controle ele está inserindo parâmetros.

Id. Sp/Pv/Ov

Nesta janela são especificadas, para o bloco de controle, as memórias em que serão obtidos os valores da variável de processo ("Vp"), do seu valor desejado ("Sp") e da variável de saída ("Ov"). Estes três parâmetros devem ser o índice de uma memória inteira (operador M) e o seu conteúdo deve estar **obrigatoriamente** na faixa do conversor, ou seja, de 0 a 4095. As variáveis "Pv" e "Sp" são variáveis de entrada para o bloco de controle, sendo lidas por ele, ao passo que a variável "Ov" é a variável de saída, sendo escrita por ele.

Id. Ganhos

Nesta janela são especificadas as memórias em que serão obtidos os valores dos ganhos do controlador PID, ou seja, ganho proporcional ("Gp"), tempo integral ("Ti") e do tempo derivativo ("Td"). Estes três parâmetros devem ser o índice de uma memória real (operador D). Estes parâmetros de controle podem ser alterados através da interface homem/máquina do MCI ou através dos painéis de controle no SPDS.

Var. Saída

Nesta janela podem ser especificados a unidade de engenharia para a variável de processo (UE Mín. e UE Máx.), o identificador da unidade de engenharia (UE Tag - máx. 4 caracteres) e os limites inferior e superior da variável de saída (OV Mín. e OV Máx.), além o valor inicial da variável (OV Ini.). Os valores relacionados à unidade de engenharia (UE Mín., UE Máx. e UE Tag) são utilizados pela supervisão para mostrar o painel de controle, não influenciando em nada no algoritmo de controle PID, ao passo que os limites da variável de saída (OV Mín. e OV Máx.) definem os limites desta variável. O valor inicial (OV Ini.) é o valor utilizado quando da carga do programa no MCI.

Faixa de Valores das Variáveis e Parâmetros do bloco PID

Os itens SP, PV e OV utilizados pelo bloco de controle PID devem estar **obrigatoriamente** em unidades do conversor, ou seja, na faixa de 0 a 4095. Se o usuário deseja converter estes valores para unidade de engenharia, deve-se fazê-lo colocando o valor em outras locações de memórias.

O ganho proporcional ("Gp") do PID deve estar na faixa entre 0 a 1000, enquanto que o tempo integral ("Ti") deve estar na faixa entre 0,001 a 5000 repetições por minuto e o tempo derivativo ("Td") na faixa entre 0 a 1000.

Tipo de Controle

Direto - indica que o controle será direto, ou seja, em uma situação de estabilidade do processo uma variação positiva da variável de processo (VP > SP) induzirá uma variação negativa na variável de saída, diminuindo-a.

Invertido - indica que o controle será invertido, ou seja, em uma situação de estabilidade do processo, uma variação positiva da variável de processo (VP > SP) induzirá uma variação positiva na variável de saída, aumentando-a.

Tipo de Saída

Direta - indica quando selecionado que o valor da saída do bloco de controle corresponderá ao máximo quando a saída OV estiver saturada no máximo.

Complementar - indica quando selecionado que o valor da saída do bloco de controle corresponderá ao mínimo quando a saída OV estiver saturada no máximo.

TAG da Malha

Nesta janela o usuário deve inserir o identificador da malha de controle correspondente a este bloco de controle. Este identificador pode ser alfanumérico e pode ter no máximo oito caracteres. Este identificador é utilizado durante a supervisão dos painéis de controle, não influenciando em nada o controle da malha.

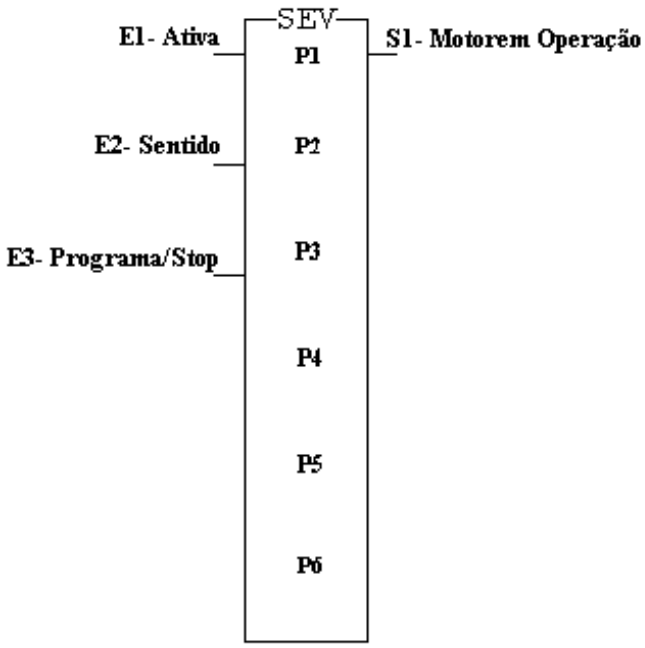
8.14.3 Controle de Servo Motor (SEV)

Objetivo - O objetivo deste elemento é interfacear servo motores com o MCI, através de um módulo de *hardware* (SCI283) do MCI02. Cada módulo pode interfacear um servo motor, permitindo programar vários parâmetros do mesmo.

Modo de Inserção do Elemento no programa

Posicionando-se o cursor na linha desejada e teclando-se a letra E. Neste instante aparecerá um menu com dezesseis opções. O usuário deve selecionar a opção “Controle de Servo Motor” e teclar “Enter”. Este elemento necessita de no mínimo seis linhas livres para a sua inserção.

Símbolo do Elemento



Parâmetros

- **P1** - representa um operador do tipo T que identifica o módulo de interface com o servo motor utilizado. Os módulos são endereçados sequencialmente, começando por T0. Assim um equipamento que possuir três placas endereçará os módulos seguintes como T1 e T2.

- **P2** - representa uma memória inteira (operador M) ou uma constante inteira (operador K) na qual é especificado o número de pulsos a serem programados no servo motor. Este valor pode variar de 1 a 65535.

- **P3** - representa uma memória inteira (operador M) ou uma constante inteira (operador K) na qual é especificado o número total de pulsos restantes programados no servo motor, em relação ao número total de pulsos programados, a partir dos quais se iniciará o procedimento de frenagem.

- **P4** - representa uma memória inteira (operador M) ou uma constante inteira (operador K) na qual é especificada a tensão da frequência de regime. Este valor pode variar de 0 a 100%.

- **P5** - representa uma memória inteira (operador M) ou uma constante inteira (operador K) na qual é especificado o índice da Taxa de Aceleração da Rampa de Subida. Este valor pode variar de 0 a 8.

- **P6** - representa uma memória inteira (operador M) ou uma constante inteira (operador K) na qual é será armazenado o valor lido do número de pulsos do *encoder* associado ao servo motor.

Entradas

- **E1** - Entrada de ativação do Servo Motor.

- **E2** - Entrada de comando que define o sentido de giro do servo motor.

- **E3** - Entrada de programação/Parada do servo motor.

Saídas

- **S1** - Saída que indica se o servo motor está ou não girando.

Descrição - Este elemento se destina a interfacear servo motores com o MCI, através de um módulo de *hardware* (SCI283) do MCI02. Cada módulo pode interfacear um servo motor, permitindo programar vários parâmetros do mesmo.

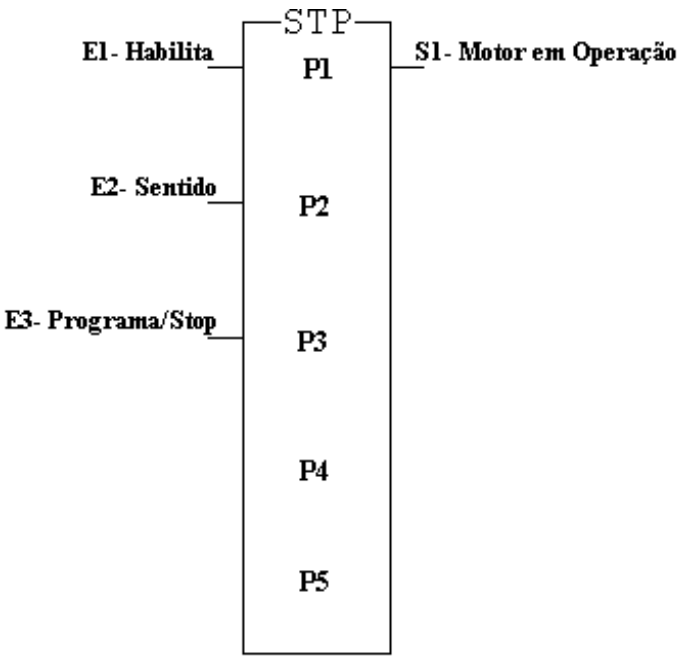
8.14.4 Controle de Motor de Passo (Step Motor)

Objetivo - O objetivo deste elemento é implementar uma interface de controle e programação de um motor de passo.

Modo de Inserção do Elemento no programa

Posicionando-se o cursor na linha desejada e teclando-se a letra E. Neste instante aparecerá um menu com dezesseis opções. O usuário deve selecionar a opção "Control*Step Motor*" e teclar "Enter". Este elemento necessita de no mínimo cinco linhas livres para a sua inserção.

Símbolo do Elemento



Parâmetros

- **P1** - representa o número de pulsos a serem programados no motor de passo. Este valor pode variar de 1 a 65535 pulsos. Este parâmetro pode ser uma memória inteira (operador M) ou uma constante inteira (operador K);

- **P2** - este parâmetro especifica a velocidade da rampa de aceleração e desaceleração do motor (Hz / ms) e pode ser uma memória inteira (operador M) ou uma constante inteira (operador K). O valor especificado representa o índice da tabela de taxa de variação da velocidade do motor, dada a seguir. São válidos para este parâmetro, valores inteiros entre 0 e 10. Qualquer valor acima deste limite, será equivalente a taxa de variação máxima, ou seja, 500 Hz/ms. Qualquer valor negativo, será equivalente ao valor mínimo, ou seja, 10 Hz/ms.

Índice	Taxa de Variação da Velocidade do Motor
0	10 Hz / ms
1	50 Hz / ms
2	100 Hz / ms
3	150 Hz / ms
4	200 Hz / ms
5	250 Hz / ms
6	300 Hz / ms
7	350 Hz / ms
8	400 Hz / ms
9	450 Hz / ms
10	500 Hz / ms

Tabela de Variação de Velocidade do Motor

- **P3** - este parâmetro é um índice de uma tabela de Frequência Inicial do motor de passo e pode ser uma memória inteira (operador M) ou uma constante inteira (operador K). Este índice pode variar de 0 a 13, sendo que para cada índice corresponde uma frequência inicial do motor. Qualquer valor acima deste limite, será equivalente a frequência máxima. Qualquer valor negativo, será equivalente ao valor mínimo. Esta tabela é representada a seguir :

Índice	Freq. Nominal	Freq. Real	Erro %
0	250 Hz	250 Hz	0
1	500 Hz	500 Hz	0
2	750 Hz	753 Hz	0.4
3	1000 Hz	1008 Hz	0.8
4	1500 Hz	1488 Hz	0.8
5	2000 Hz	2016 Hz	0.8
6	3000 Hz	2976 Hz	0.8
7	4000 Hz	3906 Hz	2.4
8	5000 Hz	4807 Hz	4.0
9	6000 Hz	6250 Hz	4.1
10	8000 Hz	7812 Hz	2.4
11	10000 Hz	10416 Hz	4.1
12	12500 Hz	12500 Hz	0
13	15000 Hz	15626 Hz	4.1

Tabela de Frequência Inicial e de Regime do Motor de Passo

- **P4** - este parâmetro é um índice de uma tabela de Frequência de Regime do motor de passo e pode ser uma memória inteira (operador M) ou uma constante inteira (operador K). Este índice pode variar de 1 a 13, sendo que para cada índice corresponde uma frequência de regime do motor. Qualquer valor acima deste limite, será equivalente a frequência máxima. Qualquer valor negativo, será equivalente ao valor mínimo. A tabela que representa a frequência de regime é a mesma mostrada anteriormente para a

frequência inicial do motor. Deve-se observar que o valor mínimo é 1 e não 0 como no caso da frequência inicial.

Entradas

- **E1**- esta entrada (Ativa *Step*), quando transitar do estado desenergizado para o energizado, ativa o motor de passo, de acordo com o número de pulsos programado pelo parâmetro P1 e no sentido definido por E2. O motor deverá, a partir deste instante, rodar de acordo com a sua programação. Para que isto aconteça, a entrada E3 deve estar desenergizada. Deve-se salientar que quando há uma transição deste sinal, apenas o número de pulsos e a direção de rotação são programados. Se o usuário desejar reprogramar a taxa de variação, frequência inicial ou frequência de regime, deve-se gerar uma transição no sinal E3.

- **E2** - esta entrada (Sentido do motor) define, através de seu estado, o sentido em que o motor deve girar. Quando energizada o motor deve girar no sentido horário, e no anti-horário caso contrário (dependente da ligação elétrica do motor de passo). Esta entrada é utilizada quando ocorre uma transição de subida do sinal E1 e o sinal E3 está inativo.

- **E3** - esta entrada (Programa/*Stop Step Motor*) quando energizada habilita a programação do step motor. Os parâmetros de programação são fornecidos através dos operadores existentes no bloco. Esta entrada tem prioridade sobre as outras entradas, ou seja, se o sinal E3 estiver ativado, independentemente do estado das entradas E1 e E2, o step motor é interrompido e reprogramado de acordo com os parâmetros do bloco. Se o motor estiver operando e a entrada E3 tornar-se ativa, o step motor é interrompido e os parâmetros em P1, P2, P3 e P4 são programados no step motor.

Saídas

- S1 - A saída do bloco reflete o estado do motor de passo, ou seja, estará energizada sempre que o motor estiver girando e desenergizada caso contrário.

Descrição - Este elemento é representado por um bloco de três entradas e uma saída, com quatro operadores.

Observação: Para que este bloco possa ser utilizado, é necessário que o MCI02 seja configurado com o módulo SSC280.

Observação: Este bloco não pode ser utilizado quando se está usando o MCI03.

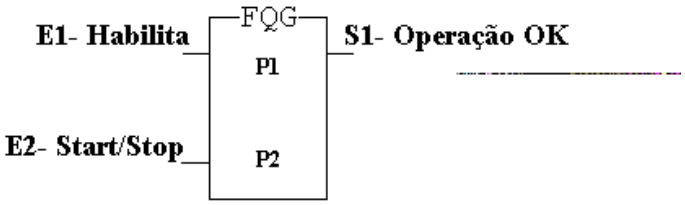
8.14.5 Gerador de Frequência (FQG)

Objetivo - Gerar uma determinada frequência através da placa geradora de frequência, que no MCI02 é o módulo SSC280. A placa geradora de frequência possui um canal de geração de frequência. Este bloco não pode ser utilizado no MCI03.

Modo de Inserção do Elemento no programa

Posicionando-se o cursor na posição desejada e teclando-se a letra E. Neste instante aparecerá um menu com dezesseis opções, entre elas, temos : “Gerador de Frequência”. O usuário deve selecionar esta opção e teclar “Enter”. Este elemento necessita de no mínimo duas linhas livres para a sua inserção.

Símbolo do Bloco



Parâmetros

- **P1** - representa o número do bloco gerador de frequência e deve ser do tipo T.
- **P2** - representa a frequência de saída da placa geradora SSC280, podendo ser uma memória inteira (operador M) ou uma constante inteira (operador K). O valor deste parâmetro pode variar de 0 a 5000, representando uma frequência de saída de 0 a 5000 Hz.

Entradas

- **E1** - entrada de habilitação. Habilita, se energizada, o tratamento da entrada **E2**. Se estiver desenergizada, não trata a entrada **E2**.
- **E2** - entrada Liga/Desliga (*Start*/*Stop*) do gerador de frequência. Se estiver energizada programa e ativa a geração da frequência de saída. Se estiver desenergizada paralisa a geração da frequência de saída.

Saídas

- **S1** - esta saída segue o estado da entrada **E1**. Se a entrada **E1** estiver energizada, a saída também estará. Caso contrário a saída estará desenergizada.

Descrição - Este elemento é representado por um bloco funcional com uma entrada, uma saída e dois parâmetros. O parâmetro **P1** deve ser um operador T, identificando a placa geradora de frequência que se deseja utilizar. O parâmetro **P2** poder ser uma constante inteira (operador K) ou uma memória inteira (operador M).

O parâmetro **P1** especifica o número do bloco gerador de frequência. O parâmetro **P2** especifica a frequência de saída do gerador.

Exemplo de Utilização

Para gerar uma frequência de 450 Hz, os parâmetros devem ser especificados como mostrado a seguir.

Operador P1	Operador P2
T0	K450

Neste exemplo será gerada, pelo bloco correspondente à saída da primeira placa geradora de frequência, uma frequência de saída de 450 Hz.

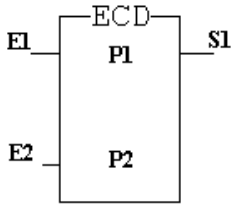
8.14.6 Interface de Encoder (ECD)

Objetivo - O objetivo deste elemento é interfacear um *encoder* com o MCI.

Modo de Inserção do Elemento no programa

Posicionando-se o cursor na linha desejada e teclando-se a letra E. Neste instante aparecerá um menu com dezesseis opções. O usuário deve selecionar a opção “Encoder” e teclar “Enter”. Este elemento necessita de no mínimo duas linhas livres para a sua inserção.

Símbolo do Elemento



Parâmetros

- **P1** - representa um canal de entrada do módulo EIM281 que servirá como entrada de contagem de pulsos do *ENCODER*. Este parâmetro deve ser um operador Txxxx onde, o valor xxxx deve coincidir com um dos três canais do módulo EIM281. Neste caso o valor de xxxx vai de 0 a 2 para o primeiro módulo EIM281, de 3 a 5 para o segundo e assim por diante.

- **P2** - representa a contagem de pulsos do *ENCODER*. Este parâmetro deve ser uma memória inteira e o seu conteúdo poderá variar de -32768 a +32767.

Entradas

- **E1** - A entrada E1 serve como habilitação do *ENCODER*, e se estiver energizada, o *ENCODER* estará habilitado a contar os pulsos e desabilitado caso contrário.

Saídas

- **S1** - Na saída S1 é gerado um pulso sempre que houver uma transição nos valores limites da contagem, que vai de -32768 a +32767, ou seja, se a contagem estiver em +32767, e o *ENCODER* fornecer mais um pulso, a contagem passará para -32768 e será gerado um pulso na saída do bloco indicando esta transição (considerando que a entrada digital de sentido está energizada - contador sendo incrementado). O mesmo ocorrerá se a contagem estiver em -32768 e o *ENCODER* fornecer mais um pulso, a contagem passará para +32767 e será gerado um pulso na saída do bloco (considerando que a entrada digital de sentido está desenergizada - contador sendo decrementado).

Descrição - Este elemento se destina à contagem de pulsos gerados por um *ENCODER*. Ele é representado por um bloco de uma entrada e uma saída, com dois operadores. A entrada do bloco serve como habilitação do *ENCODER*, e se esta entrada está energizada, o *ENCODER* estará habilitado a contar os pulsos e desabilitado caso contrário.

Na saída do bloco é gerado um pulso sempre que houver uma transição nos valores limites da contagem, que vai de -32768 a +32767, ou seja, se a contagem estiver em +32767, e o *ENCODER* fornecer mais um pulso, a contagem passará para -32768 e será gerado um pulso na saída do bloco indicando esta transição (considerando que a entrada digital de sentido está energizada - contador sendo incrementado).

O mesmo ocorrerá se a contagem estiver em -32768 e o *ENCODER* fornecer mais um pulso, a contagem passará para +32767 e será gerado um pulso na saída do bloco (considerando que a entrada digital de sentido está desenergizada - contador sendo decrementado).

Como cada módulo EIM281 possui três canais de entrada para leitura de sinais provenientes de *encoder*, os operadores T deverão ser numerados de 0 a 2, correspondendo respectivamente aos canais 0, 1 e 2 do módulo. Para o caso de ser ter mais de um módulo EIM281 no sistema, estes deverão ser numerados de forma sequencial. Se tivermos, por exemplo, dois módulos EIM281 no sistema, os canais do segundo módulo corresponderão respectivamente aos operadores T3, T4 e T5.

A frequência máxima a ser gerada pelo *encoder* não deve ultrapassar 100 KHz, valor limite superior para este elemento.

Observação: para maiores detalhes da conexão do *encoder* ao sistema, consulte a documentação de *hardware* do módulo EIM281, presente no final deste manual.

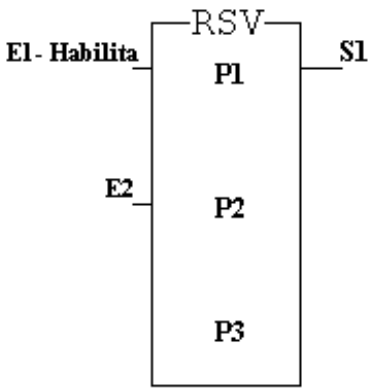
8.14.7 Interface de Resolver (RSV)

Objetivo - O objetivo deste elemento é ler a velocidade de movimentação, através de um módulo de *hardware* (EIM281) do MCI02. Este módulo pode interfacear até três *encoders* (codificadores angulares) que neste caso podem ser utilizados para ler a velocidade de giro do eixo do *encoder*.

Modo de Inserção do Elemento no programa

Posicionando-se o cursor na linha desejada e teclando-se a letra E. Neste instante aparecerá um menu com dezesseis opções. O usuário deve selecionar a opção “Interface Resolver” e teclar “Enter”. Este elemento necessita de no mínimo três linhas livres para a sua inserção.

Símbolo do Elemento



Parâmetros

- **P1** - representa um operador do tipo T que identifica o canal da placa de interface com *encoder* utilizado. Cada placa possui três canais de interface com *encoder*. Assim na primeira placa, os canais são endereçados como T0, T1 e T2, na segunda placa os canais são T3, T4 e T5 e assim por diante.

- **P2** - representa uma memória inteira (operador M) ou uma constante inteira (operador K) na qual é especificada a resolução do *encoder* usado (pulsos por volta).

- **P3** - representa uma memória inteira (operador M) na qual é atualizado a velocidade lida pelo *encoder* em rotações por minuto (rpm).

Entradas

- **E1** - Entrada de habilitação do bloco. Se estiver energizada, o bloco estará apto a operar.

- **E2** - Entrada de comando de início de leitura da velocidade. Quando houver uma transição de subida, estando ainda a entrada E1 energizada, a leitura da velocidade é iniciada.

Saídas

- **S1** - Saída que indica que o bloco está inativo ou que a leitura da velocidade está em processamento, quando desenergizada. Quando energizada indica que há uma velocidade válida no parâmetro P3.

Descrição - Este elemento se destina à leitura de velocidade de giro do eixo do *encoder* conectado ao módulo EIM281 do MCI02. O módulo EIM281 pode ser conectado a até três *encoders* simultaneamente.

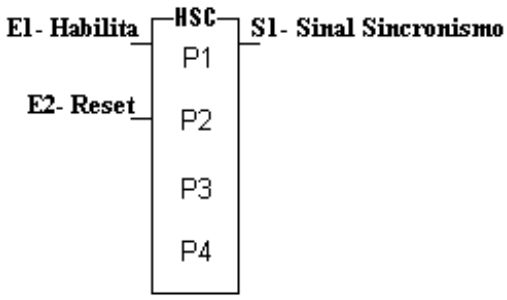
8.14.8 Contador de Alta Velocidade (HSC)

Objetivo - O objetivo deste elemento é contar pulsos, através de um módulo de hardware (EIM281/R2) do MCI02.

Modo de Inserção do Elemento no programa

Posicionando-se o cursor na linha desejada e teclando-se a letra E. Neste instante aparecerá um menu com dezesseis opções. O usuário deve selecionar a opção “Contador de Alta Velocidade” e teclar “Enter”. Este elemento necessita de no mínimo quatro linhas livres para a sua inserção.

Símbolo do Elemento



Parâmetros

- **P1** - representa um operador do tipo T que identifica o canal da placa na placa do contador rápido (EIM281/R2). Cada placa possui três canais de contagem. Assim na primeira placa, os canais são endereçados como T0, T1 e T2, na segunda placa os canais são T3, T4 e T5 e assim por diante.

- **P2** - representa uma memória inteira (operador M) na qual é atualizado o número de pulsos que estão sendo contados pelo bloco.

- **P3** - representa uma memória inteira (operador M) na qual é atualizado o número de pulsos de sincronismo do contador.

- **P4** - representa uma memória inteira (operador M) ou uma constante inteira (operador K) na qual será especificado o número de pulsos de sincronismo que a placa deverá ler para poder parar a contagem de pulsos.

Entradas

- **E1** - Entrada de habilitação do bloco contador. Se estiver energizada, o bloco estará apto a contar.

- **E2** - Entrada de *reset* do bloco contador. Quando houver uma transição de subida e este sinal permanecer ativo, estando ainda a entrada E1 energizada, o contador é programado e começará a contar os pulsos de entrada da placa EIM281/R2.

Saídas

- **S1** - Saída de indicação do sinal de sincronismo do canal. Se estiver ligada, indica que o contador está contando. Se estiver desligado, o contador não está contando.

Descrição - Este elemento se destina à contagem de pulsos através de um canal do módulo EIM281/R2 do MCI02. Opcionalmente pode-se utilizar uma entrada de sincronismo que habilita a contagem de pulsos. Se o sinal de sincronismo é utilizado, a contagem é iniciada ao ocorrer a primeira transição do sinal de sincronismo e é finalizada ao ocorrer a segunda transição do sinal de sincronismo.

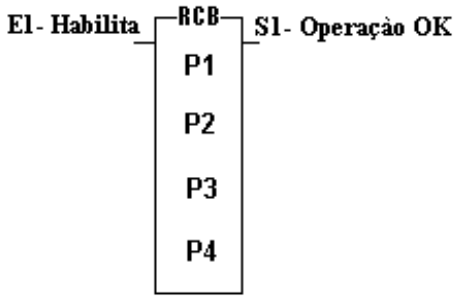
8.14.9 Comunicação com PLC Remoto (RCB)

Objetivo - O objetivo deste elemento é receber/enviar dados de/para outros controladores conectados remotamente via linha de comunicação serial.

Modo de Inserção do Elemento no programa

Posicionando-se o cursor na linha desejada e teclando-se a letra E. Neste instante aparecerá um menu com dezesseis opções. O usuário deve selecionar a opção “Comunicação com PLC Remoto” e teclar “Enter”. Este elemento necessita de no mínimo quatro linhas livres para a sua inserção.

Símbolo do Elemento



Parâmetros

- **P1** - representa uma memória inteira (operador M) a partir da qual são guardados os parâmetros dos comandos a serem executados. Estes comandos podem ser de leitura/escrita seguidos de parâmetros complementares.

- **P2** - representa um operador do tipo memória inteira (operador M), memória real (operador D) ou contato auxiliar (operador R) a partir do qual serão colocados os valores lidos remotamente ou serão retirados os valores que serão enviados para o equipamento remoto. Basicamente este operador representa o início de uma área de transferência de dados entre controladores conectados via linha de comunicação serial. O tipo do operador define implicitamente o tipo de dado que será recebido/enviado.

- **P3** - representa uma memória inteira (operador M) na qual será colocado o código de retorno da execução deste bloco.

- **P4** - representa uma constante inteira (operador K) que especifica o quantidade de operadores que serão transmitidos/recebidos através da execução deste bloco. Basicamente este parâmetro determina o tamanho da área de recepção/transmissão de dados que começa a partir do parâmetro **P2**, sendo utilizado para realizar testes de consistência da área de dados reservada pela aplicação ladder para a leitura de dados remotos.

Entradas

- **E1** - Entrada de habilitação deste bloco. Se estiver energizada, a comunicação será efetivada.

Saídas

- **S1** - Saída de indicação do resultado da execução do bloco. Se estiver ligada, indica que a função do bloco foi executada. Para saber se a função foi executada com sucesso é necessário examinar o parâmetro **P3** que contem o código de retorno.

Descrição - Este elemento se destina à comunicação entre controladores conectados via linha de comunicação serial. Pode enviar/receber dados de/para controladores remotos. A comunicação se dá através da recepção/transmissão de operadores do tipo M, D e R. O protocolo de comunicação utilizado para a comunicação é especificado através da janela de configuração de comunicação do controlador. O usuário deve especificar o tipo de protocolo desejado e que o modo de operação do canal de comunicação serial utilizado do TIPO MESTRE.

Especificação do parâmetro P1 para o protocolo SCP-HI

Para o protocolo **SCP-HI**, o parâmetro **P1** pode ser especificado como descrito a seguir: Considerando-se que o usuário especificou para este parâmetro a memória M10, teremos o seguinte:

Em M10 - deve especificar o número da estação remota a solicitar os dados (pode variar de 1 a 254)
Em M11 - deve especificar o código do comando a ser executado, podendo ser:
90h - correspondendo ao comando de Leitura de contato auxiliar (operador R), ou
94h - correspondendo ao código do comando de Escrita de contato auxiliar (operador R), ou
91h - correspondendo ao comando de Leitura de memória inteira (operador M), ou
95h - correspondendo ao comando de Escrita de memória inteira (operador M), ou
92h - correspondendo ao comando de Leitura de memória real (operador D), ou
96h - correspondendo ao comando de Escrita de memória real (operador D)
Em M12 - deve especificar o número inicial do operador que deve ser lido/escrito
Em M13 - deve especificar a quantidade de operadores que devem lidos/escritos a partir do operador especificado em M12.
Em M14 - No caso especifico do comando “Preset Multiple Registers” do protocolo ModBus - RTU, deve conter os valores que serão enviados para o controlador remoto. No caso em M14 o valor de alteração do primeiro registro, em M15 o valor de alteração do segundo registro, e assim sucessivamente. Nos comandos de escrita via protocolo HI-SCP os dados a serem enviados devem ser especificados no parâmetro P2 deste bloco.

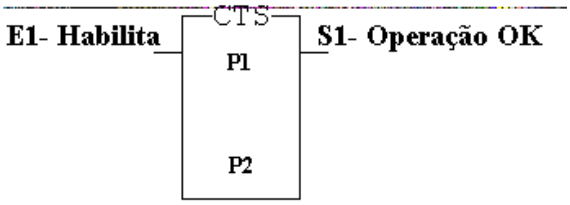
8.14.10 Configuração e Status do Controlador (CTS)

Objetivo - configurar a palavra de controle do programa *Ladder*.

Modo de Inserção do Elemento no programa

Posicionando-se o cursor na posição desejada e teclando-se a letra E. Neste instante aparecerá um menu com quinze opções, entre elas, temos : “Config./Status Contr.”. O usuário deve selecionar esta opção e teclar “Enter”. Este elemento necessita de no mínimo duas linhas livres para a sua inserção.

Símbolo do Bloco



Parâmetros

- **P1** - representa uma palavra de controle do programa Ladder. É uma palavra de 16 *bits*, onde cada *bit* representa um controle interno do programa *Ladder*. Pode ser uma memória inteira (operador M), uma constante inteira (operador K).

- **P2** - representa um parâmetro de retorno reservado para futuras funções internas do programa *Ladder* que retornem parâmetros.

Entradas

- **E1** - entrada de habilitação (Habilita) da configuração da palavra de controle do programa *Ladder*.

Saídas

- **S1** - esta saída indica o sucesso ou não da operação de configuração da palavra de controle do programa *Ladder*. Se estiver energizada, significa que a configuração com sucesso e que houve algum erro, caso esteja desenergizada.

Descrição - Este elemento é representado por um bloco funcional com uma entrada, uma saída e tem dois parâmetros. O parâmetro **P1** pode ser uma constante inteira (operador K) ou uma memória inteira (operador M). O parâmetro **P2** deve ser uma memória inteira (operador M).

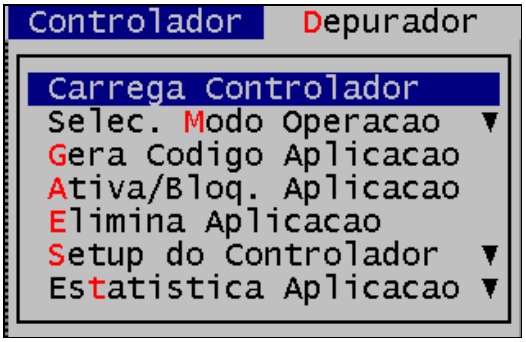
O parâmetro P1 especifica a configuração da palavra de controle do programa *Ladder*. O parâmetro P2 especifica uma memória que conterá o código de retorno reservado para futuras funções.

A saída S1 indica se a configuração da palavra de controle foi efetivada com sucesso. Se estiver energizada é porque a configuração transcorreu corretamente e desenergizada caso contrário.

Exemplo de Utilização

Atualmente foram especificados na palavra de controle somente os *bits* 0 e 1, que correspondem à habilitação de transferência de mensagens de eventos para os canais de comunicação 0 e 1. Se o *bit* 0 estiver ligado, então as mensagens de eventos que forem sendo habilitadas no programa *Ladder* serão enviadas pelo canal de comunicação 0, em formato ASCII. Se o *bit* 1 estiver ligado, então as mensagens de eventos que forem sendo habilitadas serão enviadas pelo canal de comunicação 1, em formato ASCII. As mensagens serão enviadas de acordo com a sua prioridade. Se tivermos os blocos de evento 0 e 5 ativados em um determinado instante, será enviado para a serial a mensagem correspondente ao evento 0. A mensagem correspondente ao evento 5 somente será enviada para a serial quando for desativado o bloco correspondente ao evento 0. Para o envio das mensagens de evento para a serial, valem as configurações dos canais de comunicação correspondentes.

9. Controlador



O principal objetivo desta opção é gerar código para ser executado no MCI. Adicionalmente à geração do código (opção "Gera Código Aplicação"), o mesmo deve poder ser enviado para o MCI (opção "Carrega Controlador"), via linha de comunicação serial e uma vez lá, pode-se executá-lo e também interrompê-lo (opção "Ativa/Bloq. Aplicação"). Este mesmo código, chamado de programa de aplicação, ou simplesmente Aplicação, pode ser eliminado da memória do MCI (opção "Elimina Aplicação"). Têm-se ainda a possibilidade de selecionar o modo de operação do controlador (opção "Selec. Modo Operação") e realizar uma Inicialização do MCI (opção "Setup do Controlador").

Existem dois modos de operação do MCI, que são o "Modo de Desenvolvimento" e o "Modo de Aplicação". Estes dois modos de operação são apresentados nos itens a seguir:

Modo de Desenvolvimento - É o modo utilizado para o desenvolvimento e testes de programas. Neste modo os programas de aplicação são gerados para serem carregados na memória RAM do MCI. Se o controlador for desligado, quando neste modo, o programa é perdido, sendo necessário carregá-lo novamente na memória do MCI.

Modo de Aplicação - O modo de aplicação normalmente é utilizado depois que se passou pela fase de desenvolvimento e o programa de aplicação está testado e aprovado. Neste caso o programa é gerado para rodar e permanecer gravado em uma FLASH EPROM (EPROM de gravação/leitura rápida). O MCI pode ser desligado e o programa permanece inalterado, não sendo necessário recarregá-lo. Isto somente é feito quando se deseja substituí-lo por uma versão mais atualizada.

Cada item disponível no menu "Controlador" será detalhado a seguir.

9.1 Gerar Código para a Aplicação

O código gerado será uma tradução do programa do usuário, feito no editor de diagramas de contatos, para um que o MCI possa executar. Se durante esta tradução for encontrado algum problema, este fato será informado ao usuário. O código gerado deve ser usado para enviar ao MCI onde poderá ser executado remotamente no SPDS.

O SPDS oferece ao usuário duas possibilidades para alocação do código do programa de aplicação: em RAM e em FLASH EPROM. A alocação de código para RAM é indicada para a etapa de desenvolvimento da aplicação, já que o processo de geração, carga e testes do programa fica mais rápido e fácil (disponível quando o modo de desenvolvimento estiver selecionado). Uma vez que a aplicação foi testada e aprovada em todos os testes que se fizerem necessários, é mais indicado alocar o código em memória do tipo FLASH EPROM, que ficará definitivamente rodando a aplicação.

9.2 Carrega Controlador

Uma vez que o foi gerado o código para o programa de aplicação, ele precisa ser enviado para o MCI para poder ser executado. O envio do programa para o MCI é feito via linha de comunicação serial. Ao selecionar esta opção o usuário estará comandando a execução desta operação. O fato do MCI parar de executar o programa do usuário que estiver executando para receber um outro, faz com que seja pedido uma confirmação para este comando, evitando assim quaisquer transtornos provenientes de um envio indesejado.

O código gerado é carregado em memória RAM ou FLASH EPROM do MCI, dependendo do modo de operação selecionado. Se o modo de operação for “Desenvolvimento”, o código é carregado em memória RAM e em FLASH EPROM, caso o modo de operação seja “Aplicação”.

9.3 Ativa/Bloqueia Aplicação

Uma vez que o código que foi gerado no SPDS tenha sido enviado para o MCI, ele pode ser executado remotamente através deste comando (opção "Ativa/Bloq. Aplicação"). O fato do MCI começar a executar o programa do usuário que estiver carregado em sua memória, faz com que seja pedido uma confirmação para este comando, evitando assim quaisquer transtornos provenientes de um comando indesejado. Este mesmo comando pode paralisar a execução do programa de aplicação do MCI, desde que ele esteja sendo executado.

9.4 Elimina Aplicação

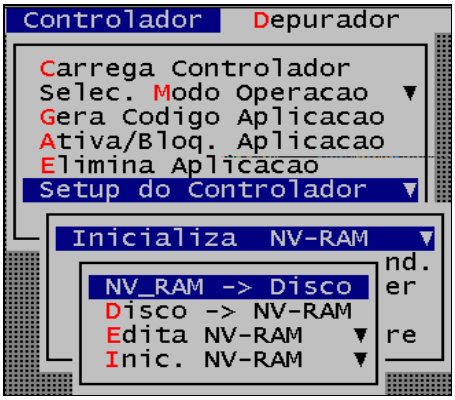
Uma vez que o código que foi gerado no SPDS tenha sido enviado para o MCI, ele pode ser eliminado remotamente através deste comando (opção "Elimina Aplicação"). Como a eliminação do programa na memória do MCI é uma tarefa importante, é necessário que ela seja confirmada, evitando assim quaisquer transtornos provenientes de um comando indesejado. É importante lembrar que o programa de aplicação é eliminado da memória do MCI, não importando o modo de operação que estiver selecionado, ou seja, não importando se o programa está em memória RAM (modo de desenvolvimento) ou FLASH EPROM (modo de aplicação).

9.5 Setup do Controlador

A opção “*Setup* do controlador” possibilita ao usuário ter acesso à vários itens relativos ao *hardware* do MCI. Através dela pode-se inicializar a memória NV-RAM, ajustar o relógio/calendário, testar o “watch dog timer”, configurar a comunicação do MCI com o SPDS e carregar um novo firmware na memória do MCI.

Estas opções serão detalhadas nos itens a seguir.

9.5.1 Inicializa NV-RAM



O SPDS permite ao usuário ter acesso à memória não volátil do MCI. Esta memória é indicada para se alocar operadores M, D e R que não percam os seus valores atuais (retentivos) quando o sistema é (re)inicializado, seja por falta de energia ou por ter-se pressionado o botão de *reset* do controlador. Usualmente nesta memória ficam os operadores que permitem configurar a aplicação do usuário, sendo reservada para parâmetros do processo que se está controlando e que se deseja que sejam programáveis. O conjunto destes operadores existentes na memória não volátil do MCI é definida aqui como base de dados não volátil, ou simplesmente "Base Não Volátil".

A opção "*Setup* do Controlador" do menu "Controlador" tem como uma das opções a "Inicializa NV-RAM", que oferece ao usuário a possibilidade de ler ("NV-RAM->Disco") a base de dados não volátil da memória do MCI e salvá-la em disco, escrever ("Disco->NV-RAM") uma base de dados existente em disco na memória do MCI, alterar os valores na base de dados que tiver sido lida do MCI e estiver presente em disco ("Edita NV-RAM") e inicializar ("Inic. NV-RAM") a base de dados não volátil na memória do MCI.

A Inicialização e edição da base não volátil no controlador, feitas através das opções "Edita NV-RAM" e "Inic. NV-RAM", pode ser feita separadamente para as bases de Contatos Auxiliares (opção "[R] Contatos Aux."), de Memórias Inteiras (opção "[M] Mem. Inteiras") e de Memórias Reais (opção "[D] Mem. Reais"). No caso da Inicialização da base de dados, pode-se executar o comando para todas as bases juntas (opção "[T] Todas Base não volátil").

9.5.2 Ajusta Relógio/Calendário

O MCI possui no módulo de CPU um relógio/calendário de tempo real (item opcional de configuração de *hardware*).

Através do SPDS, o usuário pode ter acesso ao relógio/calendário, podendo acertar a data (dia, mês e ano) e o tempo (hora, minutos e segundos).

Na opção "*Setup* do Controlador" do menu "Controlador", o usuário pode ter acesso à programação do relógio (opção "Ajusta Relógio/Calend."), podendo acertar a data e hora correntes

no relógio do MCI. Após o usuário ter entrado com novos valores para a data e hora, pode-se teclar **RET**, confirmando os novos valores para acertar o relógio ou **ESC**, cancelando a operação.

9.5.3 Testa *Watch Dog Timer*

O *hardware* do MCI possui um relógio (*watch dog timer*) que serve para indicar que o processador está funcionando. Este relógio, que na verdade é um temporizador que conta um tempo pré-programado regressivamente e que nunca deve alcançar o valor zero.

Se a contagem chegar a zero indicará que o processador teve algum problema que o impediu de reinicializar o conteúdo do temporizador antes dele zerar. Este relógio serve como um cão de guarda (em inglês: *watch dog*) do processador, sempre alerta para chamar a atenção caso o processador tenha algum problema.

Na opção "*Setup* do Controlador" do menu "Controlador", o usuário pode ter acesso à rotina de teste do "*watch dog timer*" do MCI (opção "Testa *Watch Dog Timer"),* desde que ele esteja conectado com o ambiente do SPDS.

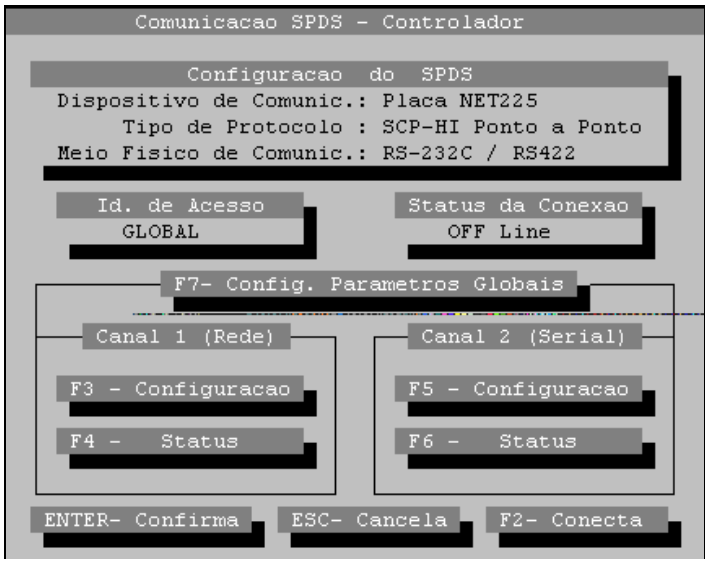
9.5.4 Configura Comunicação

Na opção "*Setup* do Controlador" do menu "Controlador", o usuário pode ter acesso à configuração da comunicação do MCI (opção "Config. Comunicação").

Através desta opção o usuário pode configurar o canal de comunicação serial que será utilizado para comunicação com o MCI (opções Serial COM1 e Serial COM2). Opcionalmente, esta comunicação pode ser feita através das placas de comunicação NET225 (fornecidas pela HI Tecnologia) configuradas para comunicação ponto a ponto (opção NET225(pp)) ou em rede (opção NET225 (rede)).

O usuário deve escolher entre uma das quatro opções disponíveis para o canal de comunicação serial. Se o usuário for utilizar a placa NET225, deve fornecer um identificador (Id-Estação) para a estação com a qual deseja comunicar (MCI).

.



O usuário têm disponível os seguintes comandos:

- **F2-Conecta**

O sistema tenta conectar com o MCI, se já não estiver conectado.

♦ **ESC - Cancela**
Para sair desta opção sem alterar os valores definidos.

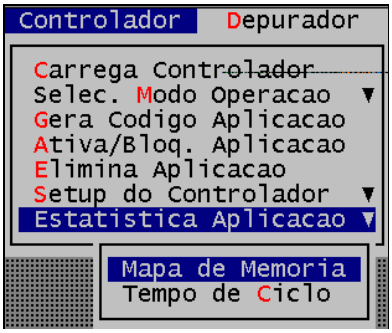
♦ **RET - Confirma**
Para confirmar as alterações desejadas.

9.5.5 Carrega Novo *Firmware*

Na opção "*Setup* do Controlador" do menu "Controlador", o usuário pode ter acesso à rotina de carregar novo *firmware* na memória do MCI (opção "Carrega Novo*Firmware*"). *Firmware* é a parte do programa do MCI que permite que ele funcione como um controlador lógico programável (CLP). Esta opção é utilizada quando se deseja atualizar o*firmware* do controlador ou a sua biblioteca de funções e deve ser utilizada apenas em situações especiais.

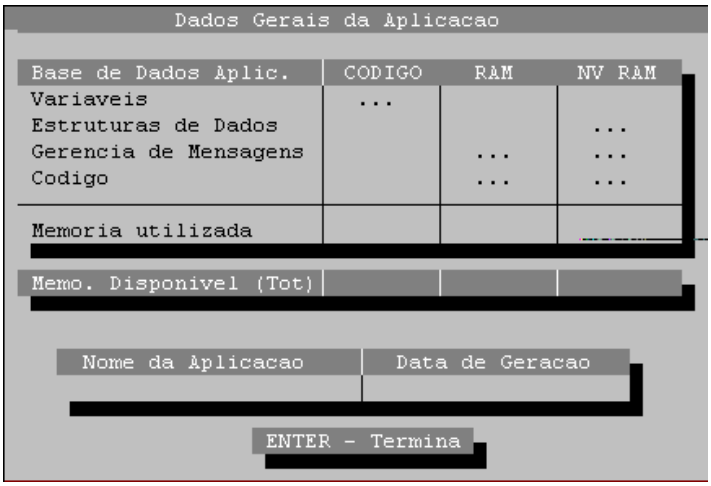
Se necessário atualizar o *firmware* do MCI, a HI Tecnologia deve entregar um novo*firmware*, em forma de arquivo, para o usuário e este deve utilizar esta opção para transmiti-lo para o MCI.

9.6 Estatística da Aplicação



A opção “Estatística Aplicação” possibilita ao usuário ter acesso dois itens relativos ao *hardware* do MCI. Através dela pode-se obter um mapa de memória do MCI (opção “Mapa de Memória”) e avaliar o tempo de ciclo do programa que estiver rodando no MCI (opção “Tempo de Ciclo”).
Estas duas opções serão detalhadas nos itens a seguir.

9.6.1 Mapa de Memória



Durante a execução de um projeto é importante manter um controle sobre os recursos ainda disponíveis e os já utilizados, uma vez que para aplicações mais complexas o usuário tem uma forte tendência de se perder frente a tantos índices e detalhes sobre a aplicação. Com o intuito de orientá-lo fornecemos um mapa detalhado da utilização da memória pela sua aplicação. Este mapa de memória apresenta a quantidade de memória utilizada pela aplicação separada em grupos, individualmente e totalizada para o caso de geração de código para ser executado em RAM e FLASH EPROM. É feita uma distinção para os três tipos de memória utilizados pelo controlador, a saber : FLASH EPROM, RAM e NVRAM (RAM não volátil).

Os grupos apresentados são :

- **Variáveis** - mostra a quantidade de memória necessária para alocar as variáveis definidas no programa do usuário.

- **Estrutura de Dados** - mostra a quantidade de memória necessária para alocar as estruturas de dados necessárias para o gerenciamento do programa do usuário.

- **Gerência de Mensagens** - mostra a quantidade de memória necessária para alocar as estruturas de gerência de mensagens utilizadas na interface homem/máquina do MCI.

- **Código** - mostra a quantidade de memória necessária para alocar o código gerado para o programa do usuário.

9.6.2 Tempo de Ciclo

Através da opção "Tempo de Ciclo" do menu "Estatística Aplicação", o usuário tem acesso a uma ferramenta de teste que permite medir o tempo de execução do seu programa no MCI. Neste caso o programa do usuário deve estar rodando no MCI.

Ao selecionar esta opção, abre-se uma tela no vídeo onde é mostrado o tempo de execução do programa do usuário no MCI. Enquanto esta tela estiver ativa, o tempo será medido continuamente e atualizado na tela. Este tipo de medição é interessante para o usuário avaliar se o seu programa está sendo executado em um período de tempo que atenda aos seus requisitos.

Na janela de "Tempos Médios de Execução", são mostrados os tempos médios de:

- "Execução do Programa" - tempo médio gasto para executar o programa do usuário (em milisegundos).

- "Gerência do Controlador" - tempo médio gasto na gerência do controlador (MCI).

- "Tempo total do Ciclo" - soma dos dois tempos anteriores.

Está também disponível para o usuário a "Performance Global" do controlador, definida como a razão percentual entre o tempo de "Execução do Programa" e o "Tempo Total do Ciclo". Esta performance dá ao usuário uma idéia de quanto tempo do ciclo de execução o seu programa está utilizando.

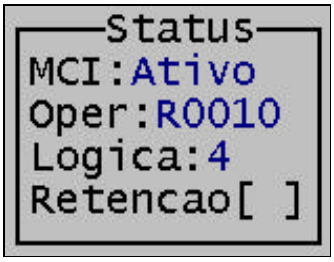
10. Depurador

Através da opção "Depurador" no menu principal do SPDS é possível ao usuário executar no MCI o programa desenvolvido e supervisioná-lo através do PC, durante a sua execução. Esta opção é bastante eficaz na procura de inconsistências no funcionamento de uma lógica, já que o usuário pode ver na tela o que está acontecendo com o seu programa no momento de sua execução.

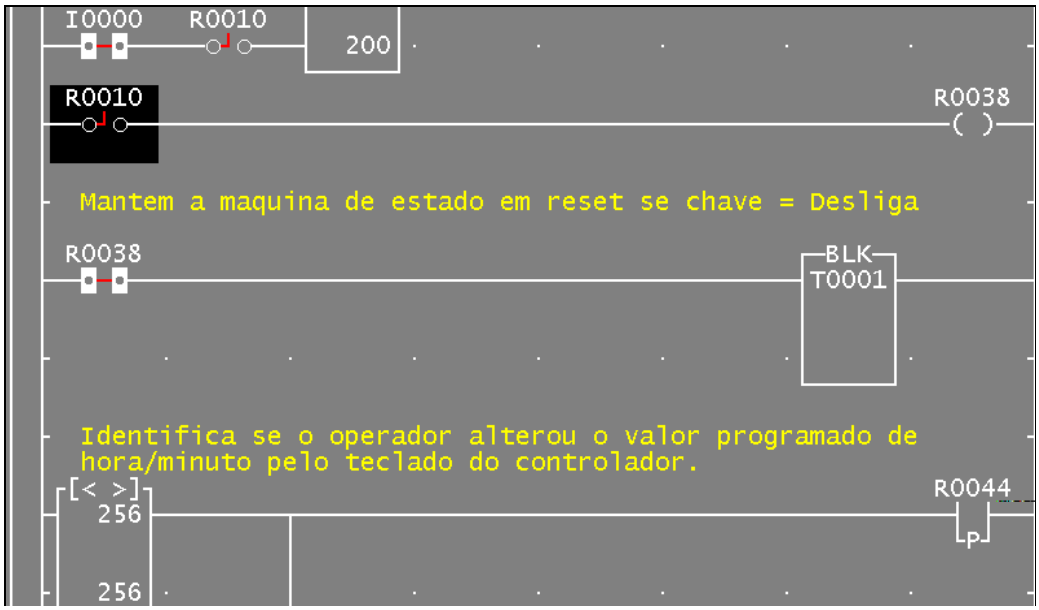
Ao selecionar a opção "Depurador" do SPDS, o sistema verifica se existe programa carregado, se existe programa no MCI, se os programas são os mesmos e se o MCI está habilitado para operar. Em caso de falha em qualquer destes testes, a opção é cancelada e uma mensagem de erro é apresentada ao usuário, caso contrário, abre-se uma tela no vídeo composta por várias sub-janelas. Cada uma delas possui um significado e função definidos como detalhado a seguir :

- **Janela de Projeto** - Tem como objetivo informar ao usuário qual o projeto corrente. É importante lembrar que todos os arquivos gerados no sistema possuem o mesmo nome do projeto, diferenciando-se apenas em sua extensão.

- **Janela de Status** - Tem como função indicar ao usuário o estado em que se encontra o MCI no instante (campo "MCI"). Este estado pode ser : Ativo ou Desligado. Esta janela também contêm informações sobre qual o operador que está sob o cursor (campo "Oper:"), informando o tipo e o seu número e se é ou não retentivo (campo "Retenção[ ]"). Têm-se qual a lógica corrente apontada pelo cursor (campo "Lógica:").



- **Janela de Depuração** - tem como objetivo mostrar ao usuário o programa que está sendo executado no MCI. Nesta janela, similar à utilizada pelo usuário para a edição do seu programa, são mostradas as lógicas definidas, com a diferença que pode-se visualizar o estado de todos os contatos e bobinas, como também o conteúdo das memórias. Existe nesta janela um cursor que ajuda na depuração do programa. Quando o cursor está sobre um elemento, aparece na janela de Status o tipo do operador, o seu número e seus valores



correspondentes. Se uma bobina está energizada, ela é mostrada com um ponto (**-(*)-**) contrastando com o seu estado desenergizado (**-( )-**). Um contato normalmente aberto é representado com o símbolo **-[ ]-**, enquanto que um contato normalmente fechado é

representado da mesma forma, só que no lugar dos colchetes são mostrados dois retângulos. Se os contatos estão abertos, eles serão mostrados com um traço simulando o contato fechado (**-[--]-)** e sem o traço (**-[ ]-)**, caso contrário. Os valores armazenados nas memórias são mostrados em todos os elementos em que uma memória aparece como operador. Durante a depuração esta janela é atualizada continuamente.

- **Janela de identificação** - tem como objetivo mostrar ao usuário qual a identificação do elemento que estiver apontado pelo cursor na janela de depuração. Este identificador é o mesmo inserido pelo usuário na etapa de programação. Esta janela fica na parte superior do vídeo e contêm o seguinte texto: "Ident:", depois do qual é mostrado o texto de identificação do operador definido pelo usuário em seu programa.

- **Janela de Comandos** - nesta janela são mostrados os comandos disponíveis para o usuário comandar a execução do programa no MCI. Estão disponíveis três comandos : "Reinicia P.", "Dump Digit." e "Dump Analog."; responsáveis pela reinicialização do programa no MCI, pela visualização e alteração das bases de dados digital e analógica, respectivamente.

- **Janela de Valor -** Nesta janela pode-se visualizar o conteúdo de qualquer operador. Quando o cursor estiver sobre qualquer operador na janela de depuração, o seu conteúdo é automaticamente mostrado nesta janela. Este campo é útil quando o valor de um operador do tipo "D" (memória real) possui um valor muito grande. Nesta situação o campo do operador no bloco correspondente indicará VMG (**V**alor **M**uito **G**rande), entretanto o valor, apesar de muito grande, poderá ser monitorado na janela de valor.

- **Janela de Identificação de Base Numérica -** nesta janela, localizada na parte superior da tela, é mostrada qual a base que esta sendo utilizada : "decimal" ou "hexadecimal".

- **Janela de Funções Numéricas do Depurador** - nesta janela são mostrados os comandos disponíveis para o depurador. Estes comandos são:

  - **F1 - *Help***
    Permite ao operador obter informações sobre os comandos e recursos disponíveis durante a depuração;

  - **F2 - Lógica**
    Permite ao usuário localizar uma determinada lógica em seu programa;

  - **F3 - Procura**
    Permite ao usuário localizar um determinado operador em seu programa;

  - **F4 - Identif.**
    Permite ao usuário alterar o texto de identificação de um operador;

  - **F5 - Docum.**
    Permite ao usuário apresentar ou não a documentação/comentário de uma lógica;

  - **F6 - Altera**
    Permite ao usuário alterar o valor do operador sob o cursor. Os operadores válidos para alteração são: M, D, R e S;

- **F7 - Force**
  Permite ao usuário inibir ou ativar uma determinada entrada digital (operador I) ou saída digital (operador O), independentemente das decisões tomadas pelo programa de controle . Quando um elemento está "forçado", o seu símbolo aparece piscando na tela do depuração.

- **F8 - Contr.**
  Permite ao usuário Ativar ou Desligar o MCI;

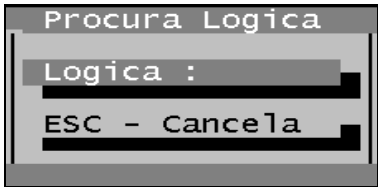
10.1 Comandos de depuração

Durante a depuração do programa, estão disponíveis para o usuário alguns comandos que visam facilitar ao máximo o teste do seu programa. Estes comandos abrangem vários recursos e serão apresentados em detalhes a seguir.

10.1.1 Auxílio ao usuário - (F1)

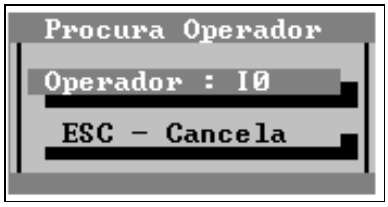
Ao se pressionar a tecla **F1**, abre-se um menu com três opções para o usuário, que são: "Comandos do Depurador", "Tipos e Variáveis", "Sintaxe" e "Blocos Funcionais". Para cada opção escolhida pela usuário, abre-se uma janela na tela, onde são apresentados os comandos e informações sobre o assunto escolhido. Este comando aparece na janela de comandos como "F1-*Help*".

10.1.2 Procura Lógica - (F2)



Ao se pressionar a tecla **F2**, abre-se uma janela que permite ao usuário especificar o número de uma lógica que deseja ver na janela de depuração. Ao ser especificada a lógica e se ela existir no programa, o depurador passará a mostrá-la e atualizá-la continuamente na tela. Este comando aparece na janela de comandos como "F2-Lógica".

10.1.3 Procura Operador (F3)



Ao se pressionar a tecla **F3**, abre-se uma janela que permite ao usuário especificar um operador que deseja encontrar no programa e mostrar na janela de depuração. Se for necessário procurar por outra ocorrência deste mesmo operador no programa, deve-se utilizar o comando "Procura próximo operador" (**ALT** e **P**) após este comando. Ao ser especificado o operador e se ele existir no programa, o depurador passará a mostrá-lo e atualizá-lo continuamente na tela. Este comando Este comando aparece na janela de comandos como "F3-Procura".

10.1.4 Procura próximo operador - (ALT P)

Ao se pressionar simultaneamente as teclas **ALT** e **P**, é procurado automaticamente a próxima ocorrência no programa do operador especificado pelo comando "Procura operador". Para procurar todas as ocorrências de um determinado operador, é só especificá-lo através do comando "Procura operador" e ir pressionando-se repetidamente as teclas **ALT** e **P**. Este comando não aparece na janela de comandos.

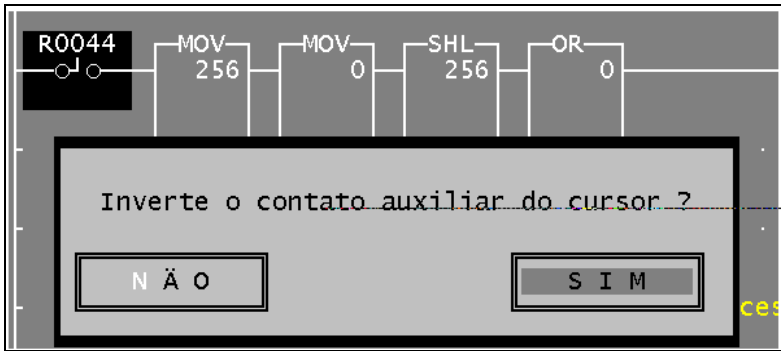
10.1.5 Consulta de Sinais - (F4)

Ao se pressionar a tecla **F4**, abre-se uma janela, onde o usuário pode escolher ver os identificadores associados aos sinais de Entrada Digital, Saída Digital, Entrada Analógica, Saída Analógica, ou os associados aos Contatos Auxiliares, Memórias Inteiras ou Reais. Este comando aparece na janela de comandos como "F4-Identif.".

10.1.6 Liga/Desliga Documentação das Lógicas (F5)

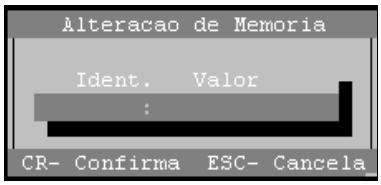
Ao se pressionar a tecla **F5** o modo de mostrar o programa do usuário na tela de depuração é alterado. Se o programa estiver sendo mostrado com a documentação/comentários das lógicas, passa a não mostrar e vice e versa.

10.1.7 Altera o valor de uma variável (F6)



É possível ao usuário alterar o conteúdo das variáveis do tipo M, D ou R presente em seu programa. Para tal é necessário que o cursor esteja posicionado em cima do operador que se deseja modificar o valor e teclar **F6**. Ao ser pressionada esta tecla, abre-se na tela uma janela que permite que o valor corrente do operador seja alterado.

No caso da tela abaixo teclou-se **F6** quando o cursor estava posicionado em cima do operador R0044, ocasionado a janela ser aberta permitindo que o seu conteúdo seja modificado. Se a resposta para a pergunta “Inverte o contato auxiliar do cursor ?” for Sim, o contato auxiliar R44 será fechado, caso contrário, permanecerá aberto, como mostrado na figura.



No caso da tela abaixo teclou-se **F6** quando o cursor estava posicionado em cima do operador M0010, ocasionado a janela ser aberta permitindo que o seu conteúdo seja modificado. Para que se altere o valor corrente de M0010 de 256 para outro valor qualquer, basta que se digite este novo valor e se pressione a tecla Return. A partir deste instante o valor digitado será enviado para a memória do MCI, substituindo o valor atual.

10.1.8 Programa máscara de retenção de sinais - (F7)

Ao se pressionar a tecla **F7**, quando o cursor estiver sobre uma bobina de saída (operador O) ou um contato de entrada (operador I), abre-se uma janela que permite ao usuário atuar na máscara de retenção de sinais do MCI. Esta atuação pode ser individual ou global.

Na atuação individual, pode-se ativar o elemento apontado pelo cursor na janela de depuração ou inibi-lo ou ainda limpar a máscara.

Ativar um elemento significa que o sinal que ele está associado será acionado, permanecendo neste estado até que o comando "Limpa máscara" seja selecionado, independente do programa de controle.

Inibir um elemento significa que o sinal que ele está associado será desligado, permanecendo neste estado até que o comando "Limpa máscara" seja selecionado, independente do programa de controle.

Limpar a máscara, significa remover todas as ativações e/ou inibições inseridas pelo usuário, deixando que o programa do usuário atue de acordo com as lógicas presentes em seu programa.

Na atuação global pode-se inibir simultaneamente todas as saídas ou limpar todos os sinais de entrada ou de saída. Este comando aparece na janela de comandos como "F7-Force", fazendo referência ao fato que ele força o estado de entradas/saídas digitais do MCI.

10.1.9 Liga/Desliga controlador - (F8)

Ao se pressionar a tecla **F8** é enviado, via linha de comunicação serial, um comando de parada para o controlador, se o mesmo estiver executando o programa do usuário, ou um comando de reinicio de execução se ele não estiver executando o programa. Este comando faz com que o programa do usuário pare de ser executado no controlador ou reinicie a sua execução. Este comando aparece na janela de comando como "F8-Contr.".

10.1.10 Sair da depuração - (ESC)

Ao se pressionar a tecla **ESC**, o programa volta ao menu principal do SPDS, deixando o modo de depuração. Este comando não aparece na janela de comandos.

10.2 Comandos para o MCI

Estão disponíveis para o usuário vários comandos para o MCI durante a fase de depuração do programa. Funcionalmente os comandos são usados para o controle do MCI e para a supervisão de variáveis. Estes comandos aparecem na janela de comandos para o MCI e são os seguintes:

Comando de Controle do MCI

- **Reinicia Programa**("Reinicia P.")
  Permite ao usuário comandar o início da execução de seu programa no MCI. Uma vez executado este comando as variáveis voláteis são reinicializadas (zeradas) juntamente com as funções de Inicialização da aplicação. Este comando é equivalente a um "*Reset*" no MCI.

Comandos de Supervisão de Variáveis do MCI

- **Supervisão de Variáveis Digitais**("Dump Digital")
  Permite ao usuário supervisionar valores de variáveis de entrada e saída digitais;

- **Supervisão de Variáveis Analógicas**("Dump Analógico")
  Permite ao usuário supervisionar e alterar valores de variáveis analógicas, memórias e contatos auxiliares;

A execução de qualquer dos comandos acima referenciados é conseguida apenas teclando-se a tecla mostrada sublinhada no texto apresentado entre parêntesis.

10.2.1 Supervisão de Variáveis Analógicas

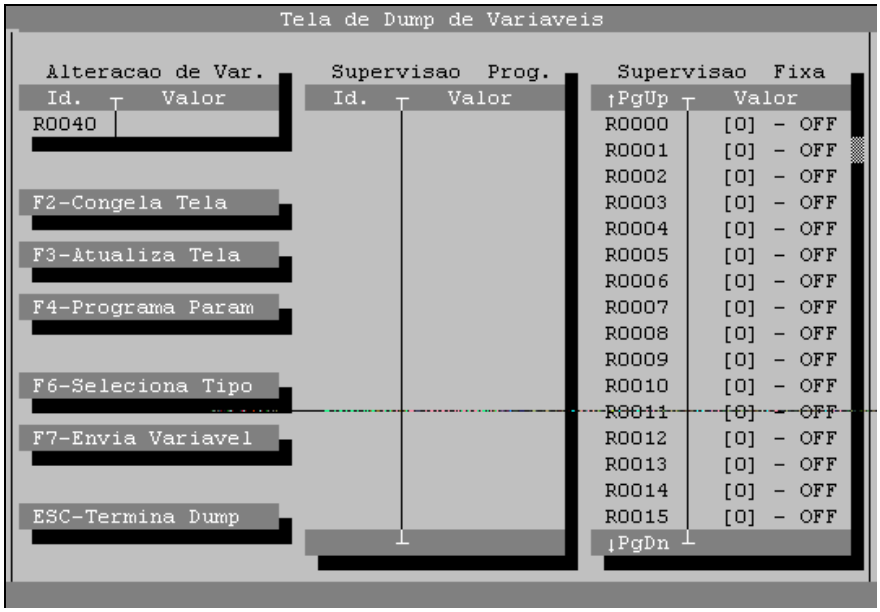
A supervisão de variáveis analógicas é conseguida através do comando "*Dump* Ana" disponível no depurador. Para selecionar este comando basta teclar-se a letra **A**.

Ao selecionar este comando, abre-se uma janela de supervisão de variáveis com três sub-janelas e alguns comandos. As sub-janelas e comandos existentes são apresentados a seguir:

- **Janela de Alteração de Variáveis**
  Esta janela é identificada em seu cabeçalho com o texto "Alteração de Var." e permite ao usuário especificar um novo valor para um operador qualquer. Para isto, inicialmente a tela deve ser congelada pelo comando **F2**. Para alterar variáveis do tipo R, digita-se 0 para desligar o contato auxiliar e 1 para ligá-lo. Nesta tela existem dois campos, no primeiro deve-se especificar o identificador do operador (campo "Id."), como por exemplo R0040 e no segundo é introduzido um novo valor para o mesmo (campo

"Valor"). Uma vez que o operador e o seu conteúdo foram especificados, pode-se enviá-lo para o MCI teclando-se **F7** (comando F7-Envia Variável), que é o comando para envio de valores para o MCI.

- **Janela de Supervisão Programada**
  Esta janela é identificada em seu cabeçalho com o texto "Supervisão Prog." e permite ao usuário introduzir o identificador de vários operadores (campo "Id.") para que o conteúdo dos mesmos sejam mostrados no campo "Valor" associado a cada um. Os operadores possíveis de especificação nesta janela são: entrada analógica (operador E), saída analógica (operador S), contato auxiliar (operador R), memória inteira (operador M) e memória real (operador D). Para inserir a lista de operadores que se deseja supervisionar nesta tela, deve-se selecionar o comando de programação "F4-Programa Param." e ao final da especificação da lista de operadores, deve-se selecionar o comando de atualização de tela "F3-Atualiza Tela" para que os valores sejam lidos e atualizados na tela.



Janelas da Tela de Supervisão de Variáveis

- **Janela de Supervisão Fixa**
  Nesta janela é possível ao usuário supervisionar todos os tipos de variáveis definidas em seu programa. As variáveis de um mesmo tipo são mostrados do índice 0 ao máximo especificado no programa do usuário. O comando **F6** ("F6-Seleciona Tipo") permite selecionar o tipo de variável que deverá ser supervisionada nesta janela. A cada vez que esta tecla for pressionada, o conteúdo desta janela é alterado para mostrar os outros tipos de variáveis. São mostradas as variáveis de entrada analógica (operador E), saída analógica (operador S), contato auxiliar (operador R), memória inteira (operador M) e memória real (operador D). Nesta janela as teclas **PgUp** e **PgDn** permitem comandar a troca de página.

Comandos de Supervisão de Variáveis

- **F2 - Congela Tela**
  Este comando faz com que a tela de supervisão de variáveis (Programada ou Fixa) tenham os valores mostrados congelados no último valor lido anteriormente ao comando e habilita o cursor para programação de variáveis a serem supervisionadas e alteração do conteúdo de variáveis.

- **F3 - Atualiza Tela**

Este comando permite ao usuário descongelar os valores das telas de supervisão (Programada ou Fixa) forçando a atualização contínua dos valores mostrados.

- **F4 - Programa Parâmetros**
  Este comando permite ao usuário especificar um operador e alterar o valor do mesmo, na janela de alteração de variáveis ("Alteração de Var.") e/ou na janela de supervisão programada ("Supervisão Prog."). Se o usuário desejar modificar o valor de uma variável, deve-se teclar **F4**, alterar o valor na janela de alteração de variáveis e a seguir teclar **F7** para enviar o valor para o MCI. Se o usuário desejar modificar a lista de variáveis na janela de supervisão programada, deve-se teclar **F4**, inserir/alterar os operadores na lista e a seguir teclar **F3** para que os valores possam ser atualizados automaticamente.

- **F6 - Seleciona Tipo**
  Este comando permite ao usuário selecionar o tipo de variável que será mostrada na janela de supervisão fixa. Podem ser selecionados os seguintes tipos de variáveis para a supervisão: variáveis de entrada analógica (operador E), saída analógica (operador S), contato auxiliar (operador R), memória inteira (operador M) e memória real (operador D).

- **F7 - Envia Variável**
  Este comando permite ao usuário enviar para o MCI o valor especificado na janela de alteração de variáveis ("Alteração de Var."). Para alterar contatos auxiliares, programe 0 para desligá-lo e 1 para ligá-lo.

- **ESC - Termina Supervisão**
  Este comando permite ao usuário terminar a supervisão de variáveis analógicas.

10.2.2 Supervisão de Variáveis Digitais

A supervisão de variáveis digitais é conseguida através do comando "*Dump* Dig" disponível no depurador. Para selecionar este comando basta teclar-se a letra **D**.

Ao se selecionar este comando, abre-se uma janela de supervisão de variáveis com duas sub-janelas e alguns comandos. As sub-janelas e comandos existentes são apresentados a seguir:

Janelas da Tela de Supervisão

- **Janela de Supervisão dos Canais de Entrada**
  Nesta janela são mostrados os canais de entrada digital. Para cada canal de entrada é fornecido o seu identificador e o seu conteúdo na base binária e hexadecimal. As teclas

Ctrl Home e **Ctrl End** permitem comandar a troca de página na tela. Os símbolos utilizados para representar os estados lógicos são os seguintes:

▪ - canal desligado (inativo)

***** - canal ligado (ativo)

L - canal desligado por retenção (inativo)

H - canal ligado por retenção (ativo)

- **Janela de Supervisão dos Canais de Saída**
  Nesta janela são mostrados os canais de saída digital. Para cada canal de saída é fornecido o seu identificador e o seu conteúdo na base binária e hexadecimal. As teclas **Ctrl PgUp** e **Ctrl PgDn** permitem comandar a troca de página na tela.

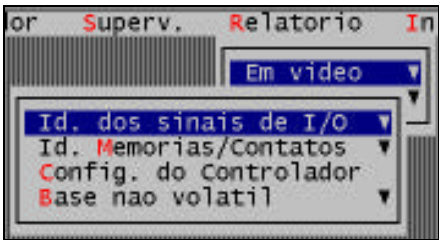
Comandos de Supervisão de Variáveis

- **F2 - Congela Tela**
  Este comando faz com que as telas de supervisão de variáveis tenham os valores mostrados congelados no último valor lido anteriormente ao comando.

- **F3 - Atualiza Tela**
  Este comando permite ao usuário descongelar os valores das telas de supervisão forçando a atualização contínua dos valores mostrados.

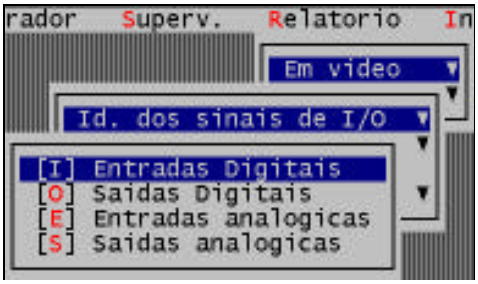
- **ESC - Termina**
  Este comando permite ao usuário terminar a supervisão de variáveis analógica.

11. Documentação



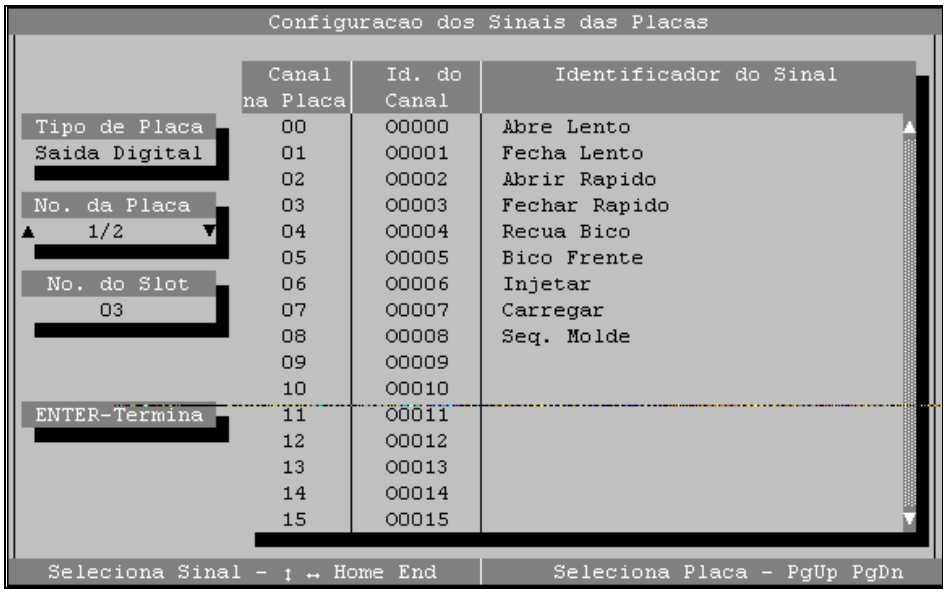
Através da opção "Relatório" do menu principal têm-se acesso ao sistema de documentação do SPDS. O relatório pode ser gerado em vídeo (opção “Em Vídeo”) ou em impressora (opção “Impressora”). Ao se selecionar a opção “Em Vídeo”, pode-se fazer uma especificação dos sinais de entrada e saída existentes no MCI, associando a cada um deles um identificador. O mesmo é válido para as memórias e contatos auxiliares utilizados no programa. O sistema de documentação fornece ao usuário a configuração completa das placas de entrada/saída do MCI. Além disto é possível gerar uma completa documentação do projeto em impressora. A seguir serão detalhados os recursos disponíveis para a opção de relatório em vídeo.

11.1 Identificação de Sinais de Entrada/Saída



Ao selecionar esta opção no menu de Documentação ("Id. dos sinais de I/O"), têm-se a possibilidade de especificar os sinais das Entradas Digitais (sub-menu "[I] Entradas Digitais"), Saídas Digitais (sub-menu "[O] Saídas Digitais"), Entradas Analógicas (sub-menu "[E] Entradas Analógicas") e Saídas Analógicas (sub-menu "[S] Saídas Analógicas").

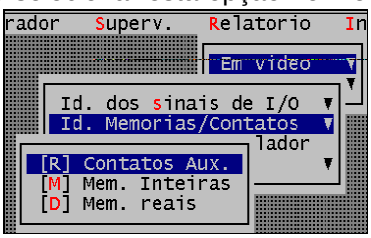
Para qualquer das opções acima citadas, abre-se uma janela de especificação de sinais. Esta janela indica o tipo de placa, se entrada/saída analógica/digital, o número da placa, o número do *slot* em que está alocada e para cada canal existente permite inserir/modificar um identificador do sinal. Deve-se lembrar que este identificador deve refletir o mais claro possível a função do sinal na aplicação.



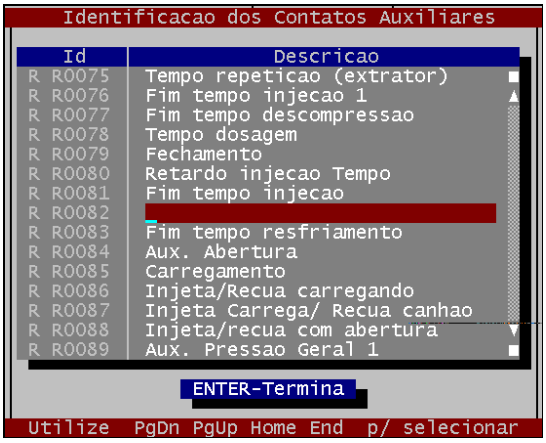
Exemplo de tela de configuração de sinais das placas (saídas digitais)

11.2 Identificação de Memórias/Contatos Auxiliares

Ao selecionar esta opção no menu de Relatório ("Id. Memórias/Contatos"), têm-se a possibilidade de especificar as memórias inteiras (sub-menu "[M] Mem. Inteiras"), memórias reais (sub-menu "[D] Mem. Reais") e contatos auxiliares (sub-menu "[R] Contatos Aux.").

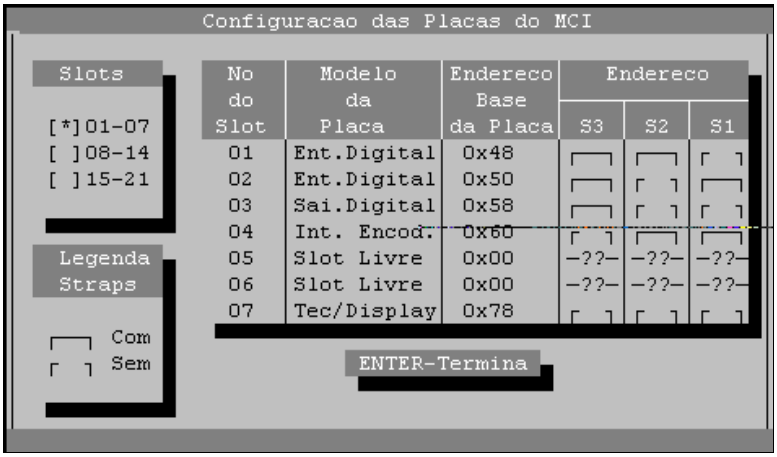


Para cada opção acima citada, abre-se uma janela de especificação de memórias/contatos auxiliares. A letra R, que aparece nesta tela, identifica uma memória/contato auxiliar Retentivo. Esta janela indica o número da memória/contato auxiliar e para cada memória/contato auxiliar existente permite inserir/modificar um identificador. Deve-se lembrar que este identificador deve refletir o mais claro possível a função da memória/contato auxiliar na aplicação.


Identificacao dos Contatos Auxiliares
Id Descricao
R R0075 Tempo repeticao (extrator)
R R0076 Fim tempo injecao 1
R R0077 Fim tempo descompressao
R R0078 Tempo dosagem
R R0079 Fechamento
R R0080 Retardo injecao Tempo
R R0081 Fim tempo injecao
R R0082
R R0083 Fim tempo resfriamento
R R0084 Aux. Abertura
R R0085 Carregamento
R R0086 Injeta/Recua carregando
R R0087 Injeta Carrega/ Recua canhao
R R0088 Injeta/recua com abertura
R R0089 Aux. Pressao Geral 1
ENTER-Termina
Utilize PgDn PgUp Home End p/ selecionar

11.3 Configuração do Controlador

A documentação gerada pelo SPDS inclui uma completa lista de configuração de todas as placas utilizadas pelo MCI02. Isto é possível através da opção "Config. do Controlador" do menu "Relatório". Ao selecionar esta opção, abre-se uma janela na tela onde pode-se ver uma especificação dos *slots* separados por grupos (00-07, 08-14, 15-21) e para o grupo selecionado, têm-se o número do *slot*, o modelo da placa que está alocada para ele, o seu endereço base e a configuração dos *straps* de endereçamento da placa. Esta configuração pode ser impressa juntamente com os outros itens da documentação.

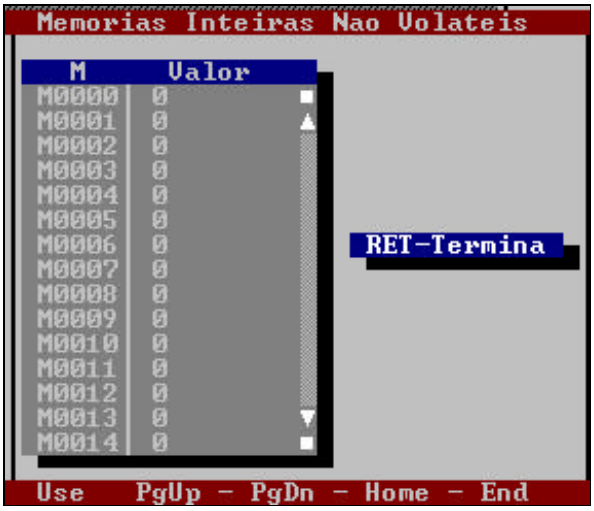


Para o caso do MCI03, em que o usuário tem que decidir entre dois tipos de módulos de I/O (MIO320 ou MIO33), não é necessária nenhuma configuração de *hardware* adicional.

11.4 Base Não Volátil

Através da opção "Base não volátil" do menu "Relatório", o usuário tem acesso ao mapa da memória não volátil do MCI. Memória não volátil é a memória que guarda os valores nela escritos mesmo quando o equipamento está desligado.

O "Mapa da Base Não Volátil" inclui os contatos auxiliares (opção "[R] Contatos Aux."), memórias inteiras (opção "[M] Mem. Inteiras") e reais (opção "[D] Mem. Reais") definidos pelo programa do usuário. Quando o usuário escolhe uma destas opções, abre-se uma janela, onde são mostrados os índices dos operadores e os seus respectivos valores, permitindo ao usuário examinar o conteúdo de qualquer elemento existente na memória não volátil.

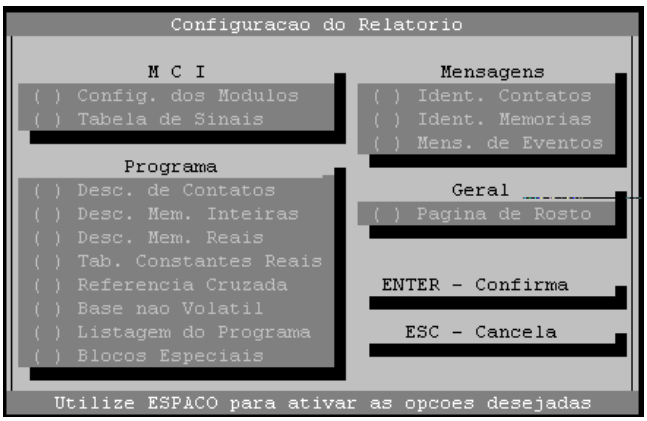


12. Impressão

Através da opção "Relatório" do menu do SPDS têm-se acesso às opções de geração de relatório em vídeo ou impressora. Ao se escolher a opção "Impressora", têm-se acesso à Configuração do documento a ser impresso (opção "Config. Documento"), da impressora (opção "Conf. Impressora") e ao recurso de impressão (opção "Imprime Documento").

Nos itens a seguir estas opções serão apresentadas em detalhes.

12.1 Impressão da Documentação


Configuracao do Relatorio
M C I
( ) Config. dos Modulos
( ) Tabela de Sinais
Mensagens
( ) Ident. Contatos
( ) Ident. Memorias
( ) Mens. de Eventos
Programa
( ) Desc. de Contatos
( ) Desc. Mem. Inteiras
( ) Desc. Mem. Reais
( ) Tab. Constantes Reais
( ) Referencia Cruzada
( ) Base nao Volatil
( ) Listagem do Programa
( ) Blocos Especiais
Geral
( ) Pagina de Rosto
ENTER - Confirma
ESC - Cancela
Utilize ESPACO para ativar as opcoes desejadas

A geração de documentação é bem versátil pois permite que o usuário escolha o tipo de informação que deseja. Antes que seja impresso ("Imprime Documento"), é apresentado na tela uma janela contendo os itens que podem constar do relatório ("Config. Documento"), se o usuário desejar pode configurar segundo a sua conveniência os dados relativos ao MCI, ao programa e se o relatório impresso conterá uma página de rosto.

.

A seguir apresentamos uma lista, separada funcionalmente, dos itens disponíveis para escolha do usuário :

- **Dados Gerais relativos ao Relatório**

  - Página de Rosto do Relatório - ao selecionar esta opção o usuário está dizendo para o sistema incluir uma página de rosto no relatório impresso.

- **Dados Relativos ao MCI**

  - Configuração dos módulos/placas do MCI - ao selecionar esta opção o usuário está dizendo para o sistema que deseja uma descrição da configuração de *hardware* do controlador. Nesta descrição são indicados os módulos de I/O utilizados pelo controlador e quais são as suas configurações, quando necessárias.

  - Tabela dos Sinais Disponíveis - ao selecionar esta opção o usuário está dizendo para o sistema que deseja uma descrição de todos os sinais disponíveis no controlador, incluindo a sua identificação.

- **Dados Relativos ao Programa**

  - Tabela de Descritores de Contatos Auxiliares - ao selecionar esta opção o usuário está dizendo para o sistema que deseja uma tabela contendo a identificação de todos os contatos auxiliares utilizados por ele em seu programa de aplicação.

  - Tabela de Descritores de Memórias Inteiras - ao selecionar esta opção o usuário está dizendo para o sistema que deseja uma tabela contendo a identificação de todos as memórias inteiras utilizadas por ele em seu programa de aplicação.

  - Tabela de Descritores de Memórias Reais - ao selecionar esta opção o usuário está dizendo para o sistema que deseja uma tabela contendo a identificação de todos as memórias reais utilizadas por ele em seu programa de aplicação.

  - Tabela de Constantes Reais - ao selecionar esta opção o usuário está dizendo para o sistema que deseja uma tabela contendo todos os valores das constantes reais definidas por ele em seu programa de aplicação.

  - Tabela de Referência Cruzada - ao selecionar esta opção o usuário está dizendo para o sistema que deseja uma tabela contendo a identificação de todos os contatos auxiliares, memórias inteiras e memórias reais utilizados por ele em seu programa de aplicação, juntamente com todas as posições (número das lógicas) do programa em que foram utilizadas.

  - Base Não Volátil - ao selecionar esta opção o usuário está dizendo para o sistema que deseja uma tabela contendo a identificação de todos os contatos auxiliares, memórias inteiras e memórias reais utilizados por ele em seu programa de aplicação, e que estiverem alocados em memória não volátil, juntamente com os respectivos valores.

  - Listagem do Programa (Diagrama *Ladder*) - ao selecionar esta opção o usuário está dizendo para o sistema que deseja uma listagem completa do programa de aplicação (diagrama *Ladder*).

  - Blocos Especiais - ao selecionar esta opção o usuário está dizendo para o sistema que deseja uma descrição de todos os parâmetros de programação especificada para os blocos especiais.

- **Mensagens relativas à IHM**

  - Identificadores de Memórias Inteiras e Reais - ao selecionar esta opção o usuário está dizendo para o sistema que deseja uma tabela contendo todas as memórias inteiras e reais utilizados por ele em seu programa de aplicação, juntamente com as mensagens definidas para a identificação do elemento na interface homem/máquina. Esta

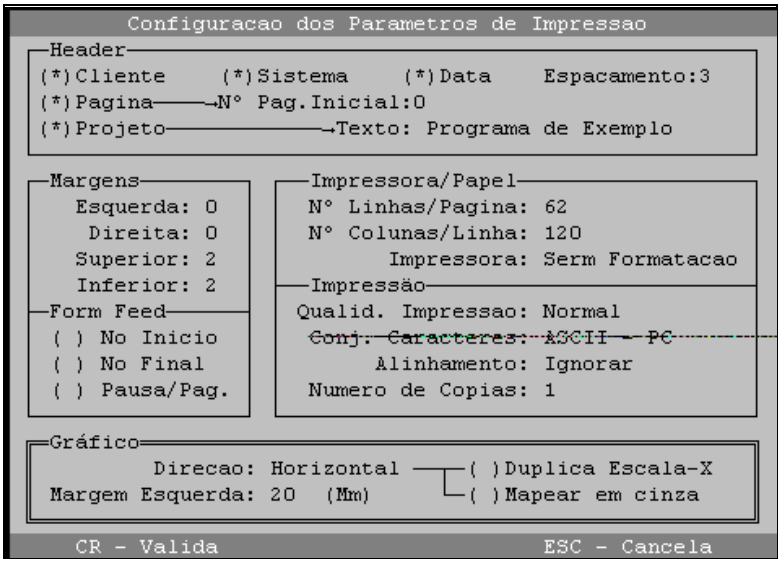
mensagem é mostrada pela interface homem/máquina quando o usuário supervisiona ou programa um destes elementos.

- Identificadores de Contatos Auxiliares - ao selecionar esta opção o usuário está dizendo para o sistema que deseja uma tabela contendo todos os contatos auxiliares utilizados por ele em seu programa de aplicação, juntamente com as mensagens definidas para a identificação do elemento na interface homem/máquina. Esta mensagem é mostrada pela interface homem/máquina quando o usuário supervisiona ou programa um destes elementos.

- Mensagens de Sinalização de Eventos - ao selecionar esta opção o usuário está dizendo para o sistema que deseja uma tabela contendo todas as mensagens definidas por ele em seu programa de aplicação. Esta mensagem é mostrada pela interface homem/máquina quando o bloco de mensagens de eventos é ativada no programa de aplicação.

Após configurar o relatório, a configuração será guardada em disco, ao se pressionar **RETURN,** ou será descartada caso se pressione **ESC**. Para imprimir o relatório, basta selecionar a opção "Imprime Documento" do menu "Impressora".

12.2 Configuração da Impressora



O SPDS permite ao usuário configurar alguns parâmetros de impressão do relatório gerado como documentação do projeto. Esta configuração é selecionada através da opção "Conf. Impressora" do menu "Impressora".

Ao ser selecionada, abre-se uma tela com várias janelas de configuração de itens do relatório. Estes itens são descritos a seguir :

Header - este item é utilizado para configurar o cabeçalho (*header*) de cada página do relatório. Nele pode-se selecionar se o cabeçalho conterá os seguintes elementos :

- Cliente - nome do usuário que está gerando a documentação.
- Sistema - nome do sistema que está sendo utilizado.
- Data - data corrente da impressão do relatório.
- Página - página corrente do relatório.
- Número da página inicial - qual deve ser o número da primeira página do relatório.
- Projeto - nome do projeto que identifica a documentação.
- Texto - Texto contendo o nome do projeto.

- Espaçamento - indica o número de linhas de espaçamento entre o cabeçalho e o texto do relatório.

Margens - este item é utilizado para configurar as margens de cada página do documento. Nele pode-se selecionar as seguintes margens :

- Esquerda - valor do número de colunas que serão saltadas à esquerda do relatório.
- Direita - valor do número de colunas que serão saltadas à direita do relatório.
- Superior - valor do número de linhas que serão saltadas no início de cada página do relatório.
- Inferior - valor do número de linhas que serão saltadas no final de cada página do relatório.

Form Feed - este item é utilizado para configurar o avanço de página do relatório. Pode-se especificar o avanço de página na seguintes situações :

- No início - a cada início de página avança-se uma página.
- No final - ao final de cada página avança-se uma página.
- Pausa/Pág. - se selecionado haverá uma pausa na impressão entre páginas.

Impressora/Papel - este item permite configurar o tipo de impressora e o tamanho do papel. a ser utilizado par emissão do relatório. Nele pode-se selecionar as seguintes margens :

- Número de linhas/página - indica o número de linhas por página desejado.
- Número de colunas/linha - indica o número de colunas por linha desejado.
- Impressora - indica qual a impressora desejada.

Impressão - este item permite configurar parâmetros de impressão do relatório. Os itens que podem ser configurados são :

- Qualidade da impressão - indica qual a qualidade da impressão desejada para o relatório. A qualidade pode ser : Normal, Condensado, Itálico e Qualidade Carta.
- Conjunto de Caracteres - indica qual o conjunto de caracteres desejado para a impressão do relatório. Este conjunto pode ser : ASCII-PC, BASCII-PC e ASCII < 127.
- Alinhamento - indica qual o alinhamento desejado para a impressão do relatório. Este alinhamento pode ser : Esquerda, Direita, Central e Ignorar.
- Número de Cópias - indica o número de cópias do relatório que se deseja imprimir.

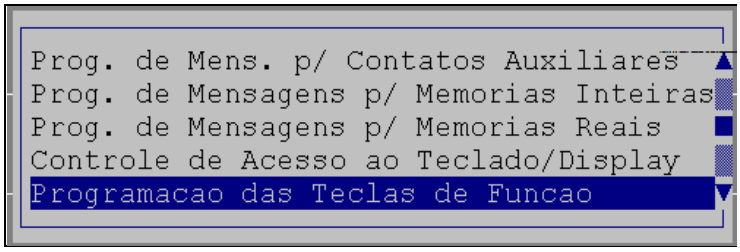
Uma vez feita a configuração da impressora, esta permanecerá válida para todos os relatórios emitidos, até que se altere algum dos itens descritos acima.

13. Interface Homem Máquina

13.1.1 Configuração da função associada a uma tecla

As teclas, de **F1** a **F6** ,as numéricas de **0** a **9**, a tecla . e +/–, podem ter associadas funções que permitem:

- Ver e/ou Modificar uma memória inteira (M)
- Ver e/ou Modificar uma memória real (D)
- Ver e/ou Modificar um contato auxiliar (R)
- Ligar um contato auxiliar (R)
- Desligar um contato auxiliar (R)
- Inverter um contato auxiliar (R)


Prog. de Mens. p/ Contatos Auxiliares
Prog. de Mensagens p/ Memorias Inteiras
Prog. de Mensagens p/ Memorias Reais
Controle de Acesso ao Teclado/Display
Programacao das Teclas de Funcao

Para se associar uma função à tecla é necessário que se entre na tela de configuração. Para isto é necessário estar editando um programa e teclar-se a combinação das teclas “ALT” + “H”. Quando estas teclas são pressionadas, aparece na tela do editor o seguinte menu:

Deve-se selecionar a opção “Programação das Teclas de Função” que permitirá configurar as teclas de função do teclado. Ao se pressionar “enter”, aparecerá na tela a seguinte janela:

Figura - Tela de configuração de Teclas de Função

Para se associar uma função a uma tecla, deve-se selecionar inicialmente a tecla. Através das teclas de seta chega-se à tecla desejada, e com auxílio da tecla de espaço, escolhe-se a função desejada que pode ser:

- R: Modifica um contato
- M: Modifica uma memória inteira (M)
- D: Modifica uma memória real (D)
- L: Liga um contato (R)
- E: Desliga um contato (R)
- I: Inverte um contato (R)
- X: sem função associada

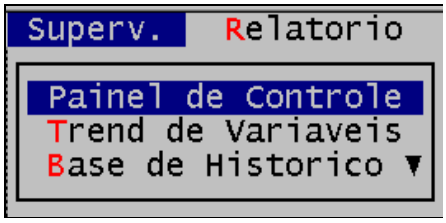
Após a escolha da função, deve-se especificar o contato, memória real ou inteira que será modificada com o auxílio da tecla de função. Isto pode ser feito com o auxílio das teclas numéricas.

Como exemplo, a figura anterior, mostra a tela de configuração após o usuário ter programado as teclas **F1**, **F2**, **F3**, **F4**, **F5**, **F6**, e **+/-**.

Neste caso, a teclas foram configuradas da seguinte maneira:

- **F1** - configurada para permitir alterar o estado do contato R5 quando esta tecla for pressionada.
- **F2** - configurada para permitir alterar o conteúdo da memória inteira M10 quando esta tecla for pressionada.
- **F3** - configurada para permitir alterar o conteúdo da memória real D15 quando esta tecla for pressionada.
- **F4** - configurada para ligar o contato R1 quando esta tecla for pressionada.
- **F5** - configurada para desligar o contato R2 quando esta tecla for pressionada.
- **F6** - configurada para inverter o contato R3 quando esta tecla for pressionada. Inverter o contato significa que se o contato estiver desligado, ele será ligado e se estiver ligado, será desligado.
- **+/-** - configurada para inverter o contato R4 quando esta tecla for pressionada. Inverter o contato significa que se o contato estiver desligado, ele será ligado e se estiver ligado, será desligado.
- As outras teclas permaneceram sem funções associadas.

14. Supervisão



Um dos recursos que o SPDS oferece ao usuário é a supervisão de Blocos de Controle e de Variáveis Analógicas. Para que se possa utilizar este recurso, basta selecionar no menu principal a opção "Superv." e através dela pode-se escolher entre "Painel de Controle" e "*Trend* de Variáveis" (gráfico de tendência de variáveis) e "Base de Histórico". A supervisão é feita durante a execução do programa do usuário no MCI. Para que ela se realize é necessário que o Ambiente de Desenvolvimento de Programas esteja conectado ao MCI e que o programa que estiver sendo rodado no MCI coincida com o existente no projeto selecionado. A seguir são apresentadas estas opções em detalhes.

14.1 Painel de Controle

A supervisão de Blocos de Controle do SPDS permite que se especifique a supervisão simultânea de até quatro blocos de controle "PID", através de um painel de supervisão. Para tanto abre-se uma janela onde têm-se quatro janelas que permitem ao usuário ativar um painel (painéis número 1, 2, 3 e 4) e programar a sinalização de alarme para o painel.

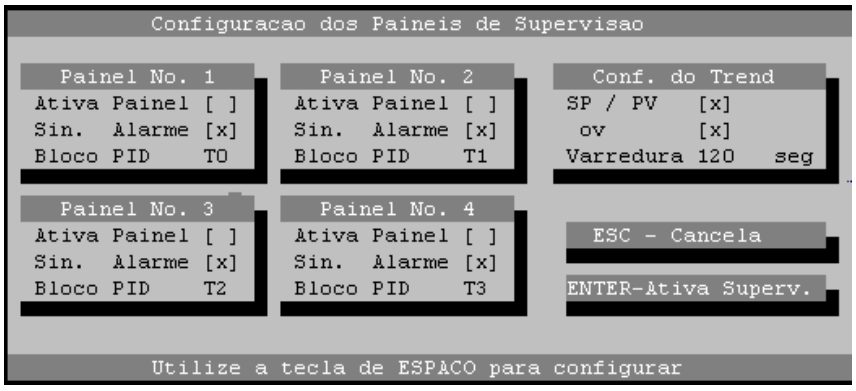


Figura Janela de configuração dos painéis de supervisão de blocos PID

Os três campos existentes na janela de programação do painel podem ser assim definidos:

- **"Ativa Painel" -** se ativado, o painel correspondente será mostrado na tela durante a supervisão.

- **"Sin. Alarme" -** se ativado, será sinalizado no painel correspondente a ocorrência de uma situação de alarme. Esta sinalização ocorre para duas situações: a variável de processo está menor do que o limite inferior de alarme ou a variável de processo está maior do que o limite superior de alarme. Estes limites são programados, no editor de diagramas *Ladder,* colocando-se o cursor sobre o bloco de controle e teclando-se **ALT** e **T**. É importante salientar que estes valores de alarme somente são válidos para a supervisão, para efeito de sinalização ao usuário, através dos painéis de controle, não influindo em nada no controle do processo.

- **"Bloco PID" -** indica qual o bloco de controle PID que estará associado ao painel de controle (T0, T1, T2 e T3).

Durante a supervisão, o usuário têm a possibilidade de ver graficamente o comportamento do controle do seu processo, através de um gráfico de tendência. São apresentados dois gráficos de tendência, sendo um mostrando a variável de processo e o seu valor desejado (*setpoint*) em função

do tempo e um outro mostrando a variável de atuação em função do tempo. Nestes gráficos, são mostradas as variações dos sinais (variáveis) em função do tempo. Quando o gráfico alcança o final da escala do tempo, o mesmo é refeito, recomeçando a mostrar os valores a partir da metade da escala de tempo, ou seja, quando o gráfico está totalmente preenchido e vai ser refeito, pega-se a metade final dos valores mostrados e os coloca no início da escala, passando a mostrar os novos valores a partir deste ponto.

Na janela de configuração do gráfico de tendência ("Conf. do *Trend*") o usuário tem os seguintes campos:

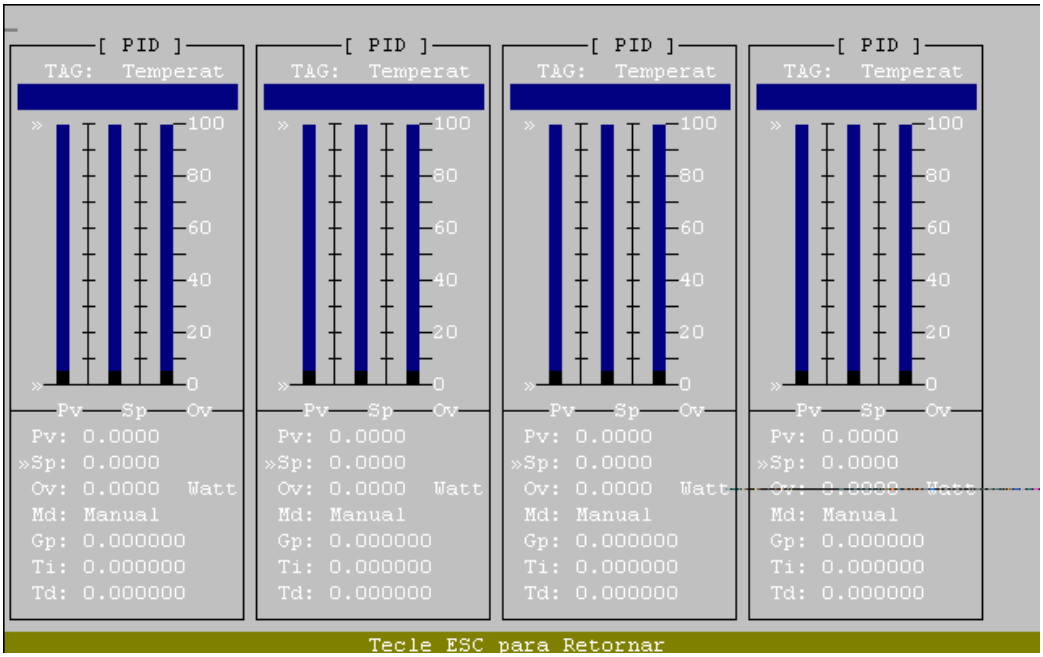


Figura - Exemplo de painéis de supervisão de blocos de controle PID

- **"SP/PV" -** se ativado, serão mostrados no gráfico de tendência a variável de processo "PV" e o seu valor desejado "SP" (*setpoint*). O usuário não tem acesso a este campo, indicando que serão sempre mostradas estas duas variáveis em um mesmo gráfico.

- **"OV"-** se ativado, será mostrado no gráfico de tendência a variável de atuação no processo "OV". O usuário não tem acesso a este campo, indicando que será sempre mostrada esta variável em um gráfico.

- **"Varredura"-** especifica a escala de tempo do gráfico de tendência durante a supervisão, representando o tempo que será gasto para ser preenchida a tela. Este valor pode variar de 100 segundos a 9999 segundos.

Uma vez feita a configuração, pode-se supervisionar os blocos de controle. Individualmente, nos painéis de supervisão dos blocos de controle são mostrados os seguintes itens :

- ***TAG*** (campo "*TAG*") **-** identificação do bloco de controle, especificado pelo usuário no editor de diagramas *Ladder* ;

- **Modo de Operação** (campo "Md") **-** pode ser manual ou automático. Este valor é controlado pela entrada de modo de operação do bloco de controle PID;
- **Variável de Processo** (campo "Vp") - é mostrado o seu valor numérico, em unidade de engenharia e em diagrama de barras ("*bargraph*"). São mostrados também os limites superior e inferior de alarme (pequenos triângulos deitados), que podem aparecer piscando caso a variável de processo ultrapasse os seus limites.

- **Valor Desejado** (Campo "Sp") **-** é mostrado o seu valor numérico, em unidade de engenharia e em diagrama de barras ('*bargraph*').

- **Variável de Saída** (campo "Ov") **-** é mostrado o seu valor numérico, em unidade de engenharia e em diagrama de barras ('*bargraph*').

- **Ganho Proporcional** (campo "Gp") **-** é mostrado o ganho proporcional do controlador PID.

- **Tempo Integral** (campo "Ti") **-** é mostrado o tempo integral do controlador PID.

- **Tempo Derivativo** (campo "Td") **-** é mostrado o tempo derivativo do controlador PID.

- **Limite Superior de Alarme** (triângulo deitado) - este limite é mostrado ao lado do "*bargraph"* da variável de processo e indica o limite máximo que a variável de processo pode assumir sem que haja uma situação de alarme. Quando a variável de processo ultrapassa este limite e se a sinalização de alarme foi habilitada, soará uma sirene e este limite passará a piscar, indicando que este foi o limite ultrapassado. O usuário deve reconhecer o alarme, através do comando "F4-*Ack_Alm*", resolver o problema que ocasionou o mesmo e quando o sistema voltar ao normal, habilitar novamente a sinalização de alarme, através do comando "F5-*Rst_Alm*".

- **Limite Inferior de Alarme** (triângulo deitado) - este limite é mostrado ao lado do "*bargraph*" da variável de processo e indica o limite mínimo que a variável de processo pode assumir sem que haja uma situação de alarme. Quando a variável de processo ultrapassa este limite e se a sinalização de alarme foi habilitada, soará uma sirene e este limite passará a piscar, indicando que este foi o limite ultrapassado. O usuário deve reconhecer o alarme, através do comando "F4-*Ack_Alm*", resolver o problema que ocasionou o mesmo e quando o sistema voltar ao normal, habilitar novamente a sinalização de alarme, através do comando "F5-*Rst_Alm*".

Os comandos disponíveis para o usuário durante a supervisão dos blocos de controle são:

- **F1-Aux. -** Permite ao usuário obter informações de auxílio sobre a operação do sistema;

- ← → **Painel -** permite selecionar um painel, entre os que estão sendo mostrados na tela. O painel que estiver selecionado, é mostrado com uma borda dupla e os comandos **F3**, **F4** e **F5**, quando selecionados, se referem ao painel selecionado.

- ↑ ↓ **F2-Param. -** Permite selecionar em um painel, qual o parâmetro que o usuário deseja modificar, podendo ser : "Sp", "Ov", "Gp", "Ti" e "Td". Para modificar um dos parâmetros selecionados, basta teclar **F2** que o sistema permitirá ao usuário modificar o seu valor. Se a modificação for confirmada pelo usuário, o parâmetro será enviado para o MCI.

- **F3-*Trend* -** Permite ao usuário passar a supervisionar o processo através de gráficos de tendência.

- **F4 - *Ack_Alm* -** Permite ao usuário reconhecer uma sinalização de alarme no painel que estiver selecionado. Quando a sirene tocar, indicando a ocorrência de um alarme no painel, o usuário deve reconhecer o alarme, indicando para o SPDS que teve conhecimento da ocorrência e o autorizando a desligar a sirene. O reconhecimento do alarme pode ser feito teclando-se a tecla **F4**.

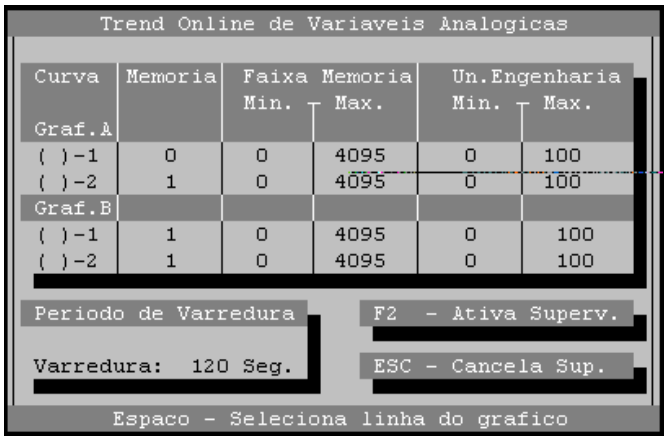
- **F5-*Rst_Alm* -** Uma vez que houve uma situação de alarme e esta foi reconhecida pelo comando "F4-*Ack_Alm*", para que o sistema volte a sinalizar outras ocorrências é necessário habilitar novamente o sistema de alarme. Isto é feito através deste

comando. Se ele não for executado, o sistema não mais sinalizará qualquer ocorrência de alarme.

- **ESC-Fim -** Permite ao usuário sair da supervisão do processo.

O sistema permanecerá em supervisão até que o usuário pressione a tecla "ESC-Fim".

14.2 *Trend* de variáveis



O "*Trend* de Variáveis" (gráfico de tendência) de Variáveis Analógicas do SPDS permite que se especifique a supervisão simultânea de até quatro variáveis, duas a duas. Para tanto abre-se uma janela onde têm-se um campo de ativação da supervisão de até duas variáveis por gráfico em dois gráficos (Graf-A e Graf-B). Associado a eles existe o número da memória onde se encontra o valor analógico a ser supervisionado, os seus valores mínimos e máximos e a sua unidade de engenharia (valores mínimos e máximos). Pode-se também especificar a escala máxima de tempo dos gráficos, variando de 100 segundos a 9999 segundos. Uma vez feita a configuração, pode-se supervisionar as variáveis (memórias).

A tela de supervisão de variáveis mostra uma ou duas janelas com gráficos de tendência das variáveis, dependendo da configuração da supervisão. Em cada janela mostrada pode-se ter uma ou duas variáveis sendo mostradas simultaneamente, o que também depende da configuração.

Para ativar a supervisão o usuário deve teclar **F2** ou **ESC** para cancelá-la. Se quiser imprimir o gráfico de tendência, deve pressionar a tecla **F5** que imprime um *Hardcopy* da tela (cópia do conteúdo da tela em impressora).

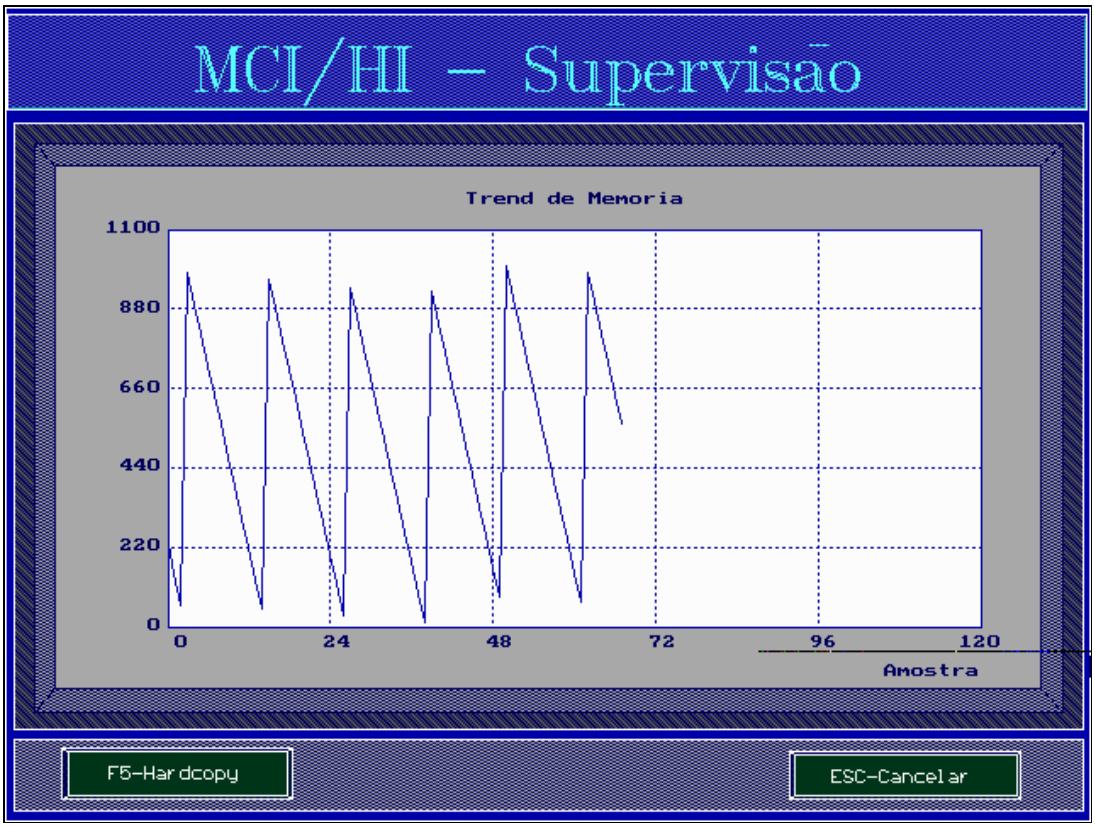
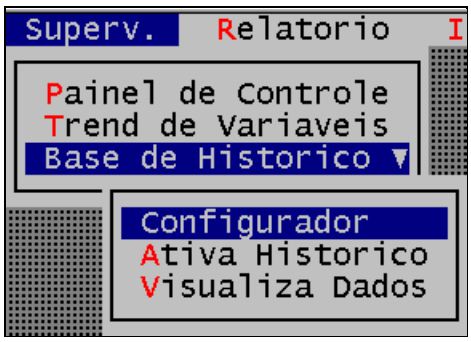


Figura - Exemplo de um gráfico de tendência com uma variável do MCI

14.3 Base de Histórico



A opção "Base de Histórico" do SPDS permite que se crie uma base de dados de até quatro variáveis lidas da memória do MCI. A opção “Configurador” permite especificar a base de dados, bem como o nome do arquivo que a conterá. A opção “Ativa Histórico” permite iniciar a aquisição de dados da memória do MCI e gravá-la em um arquivo em disco. A opção “Visualiza Dados” permite visualizar os dados adquiridos em forma de gráfico.

Ao selecionar a opção “Configurador” , abre-se uma janela na qual deve-se especificar o nome do arquivo que conterá a base de dados.

Figura - Janela de especificação do nome do arquivo que conterá o histórico

Após especificar o nome do arquivo, abre-se uma janela de configuração na qual pode-se especificar vários itens, como relacionado abaixo:

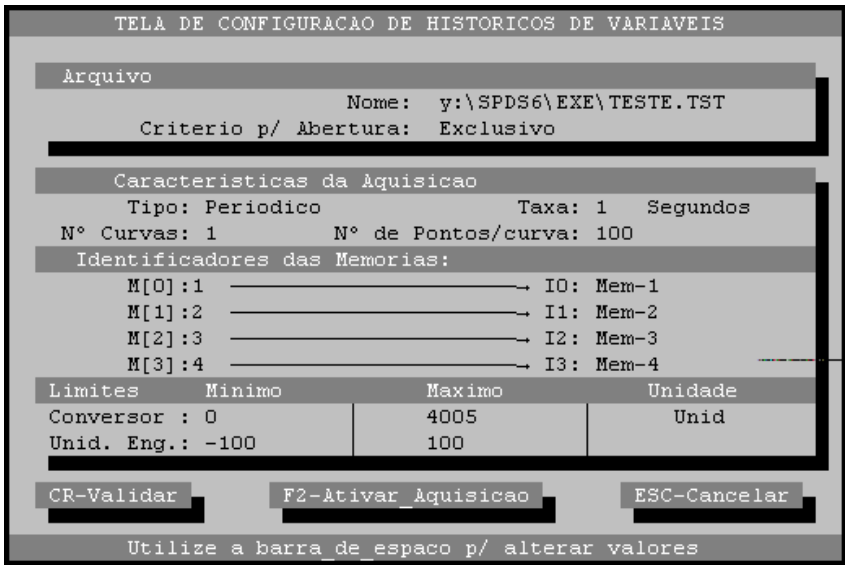


Figura - Tela de configuração de históricos de variáveis

Critério p/ Abertura do arquivo - Este parâmetro especifica o critério para abertura do arquivo. Podendo ser Exclusivo, Apendar e Sequencial. Estas opções são detalhadas a seguir:

Exclusivo - se o critério escolhido for exclusivo, será criado um novo arquivo com o nome especificado na janela de especificação. Se existir um arquivo com o mesmo nome, ele será eliminado.

Apendar - se o critério escolhido for apendar, será aberto o arquivo com o nome especificado na janela de especificação e os dados serão adicionados ao arquivo após o último dado gravado.

Sequencial - se o critério escolhido for sequencial, será criado um novo arquivo com o nome especificado na janela de especificação, mas com a extensão ".001". Toda vez que for ativada uma nova aquisição de dados (opção "Ativa Histórico"), será criado um novo arquivo com o mesmo nome do anterior, mas com a extensão incrementada de um. Ao se executar várias aquisições, as extensões dos arquivos serão: ".001", ".002", ".003" e assim por diante.

Tipo da aquisição de dados - Este parâmetro especifica o tipo da aquisição de dados, podendo ser "Periódica" ou "Em blocos". Estas opção são detalhadas a seguir:

Periódica - se o tipo de aquisição escolhido for periódica, a aquisição se dará periodicamente, sendo adquiridos os dados das memórias em intervalos de tempos pré-determinados e especificados através do parâmetro Taxa.

Em bloco - se o tipo de aquisição escolhido for em bloco, a aquisição se dará de uma única vez, sendo adquiridos todos os dados especificados.

Taxa de aquisição de dados - Este parâmetro especifica o intervalo de tempo entre as aquisições de dados. Seu valor é dado em segundos.

Número de curvas - Este parâmetro especifica o número de curvas que serão traçadas na visualização dos dados. Este valor pode variar de 1 a 4 curvas.

Número de pontos por curva - Este parâmetro especifica o número de aquisições que deverão ser realizadas por curva.

Unidade de Engenharia - Os parâmetros relacionados à unidade de engenharia permitem especificar os limites mínimos e máximos para os valores que serão mostrados no gráfico e qual a sua relação com o valor real das variáveis (conversor) e também qual a unidade de engenharia correspondente.

Ao se especificar a configuração, deve-se iniciar a aquisição dos dados. Isto é feito através da opção “Ativa Histórico” do menu “Base de Histórico”. Ao iniciar a aquisição é aberta uma janela no vídeo que mostra o nome do arquivo que está sendo criado, o andamento da aquisição e as horas inicial, final e atual da aquisição. Nesta tela é possível comandar a suspensão da aquisição a qualquer instante.

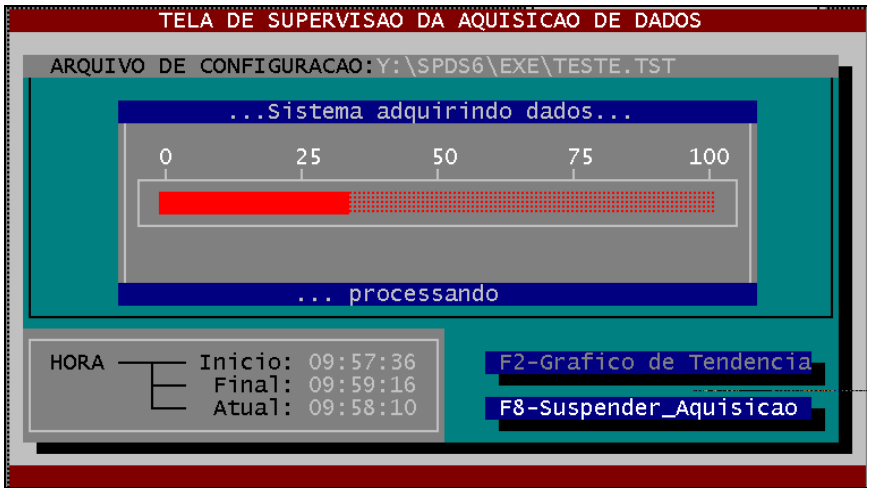


Figura - Tela de supervisão da aquisição de dados do histórico das variáveis

]Através da opção “Visualiza Dados” do menu “Base de Histórico”, pode-se visualizar os dados adquiridos em forma gráfica, conforme especificado através da opção “Configurador” do menu “Base de Histórico”.

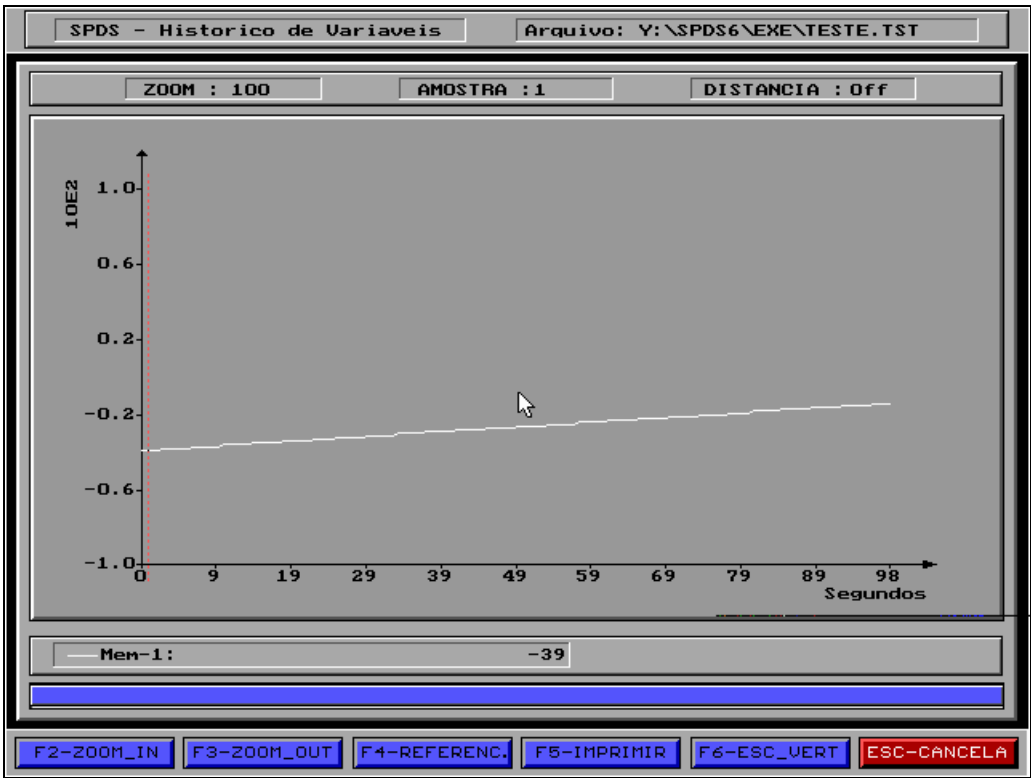
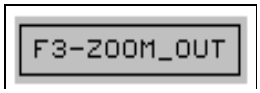


Figura - Tela de exibição de histórico de variáveis

Na janela de visualização dos dados existe um menu, associado a teclas de função F2, F3, F4, F5 e F6 e uma associada à tecla de*Escape* (ESC). Estas teclas e as funções que desempenham são apresentadas a seguir:

Esta opção permite visualizar uma parte do gráfico com mais detalhes

Esta opção permite visualizar uma parte do gráfico com menos detalhes



Esta opção permite fazer medições no gráfico. Com ela pode-se marcar uma referência no eixo do tempo e adicionalmente, com ajuda de um cursor, marcar outro ponto no eixo do tempo, permitindo medir o tempo decorrido entre as duas aquisições marcadas.

Esta opção permite imprimir o gráfico mostrado na tela.

Esta opção permite alterar a escala vertical. É possível alterá-la dos valores padrões até os valores máximo e mínimo.

Esta opção permite sair da visualização dos dados.